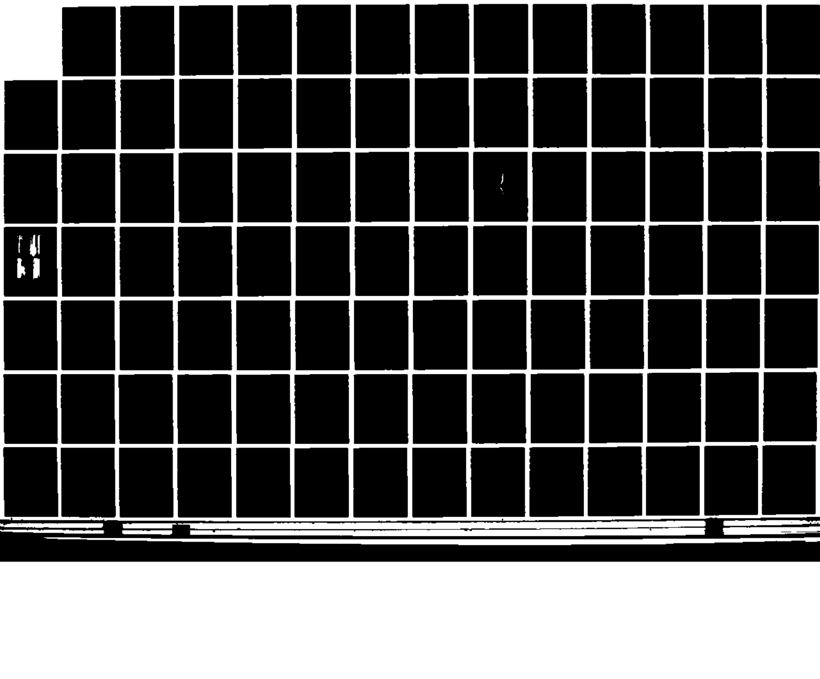
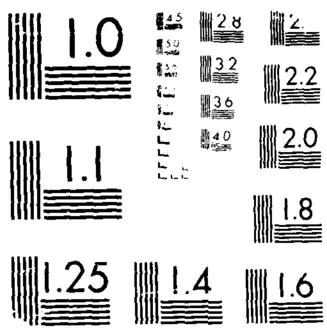
AD-A152 952 A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF 1/3
NOTCHED COMPOSITE..(U) DREXEL INST OF TECH PHILADELPHIA
PA DEPT OF MECHANICAL ENGINE.. A S WANG ET AL. FEB 85

UNCLASSIFIED AFOSR-TR-85-0280 AFOSR-84-0334 F/G 11/4 NL





MICROCOPY RESOLUTION TEST CHART

DTIC FILE COPY

AFOSR-TR- 88-0280

A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF NOTCHED COMPOSITE LAMINATES

AD-A192 952

A. S. D. Wang, E. S. Reddy, W. Binienda and Y. Zhong

Department of Mechanical Engineering
Drexel University
Philadelphia, PA. 19104

FINAL TECHNICAL REPORT

Grant No. AFOSR-84-0334

30 September 1984 to 31 December 1987

for

Director of Aerospace Sciences

Air Force Office of Scientific Research

Bolling Air Force Base, DC 20332

February, 1988

UNCLASSIFIED ADA192952

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT A **Approved for public release, distribution unlimited**
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 88-0280
6a. NAME OF PERFORMING ORGANIZATION Drexel University; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFOSR/NA
6c. ADDRESS (City, State, and ZIP Code) 32nd and Chestnut Sts. Philadelphia, Pa. 19104	7b. ADDRESS (City, State, and ZIP Code) Build 410 Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR/NA; 8b. OFFICE SYMBOL (If applicable) NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-84-0334
8c. ADDRESS (City, State, and ZIP Code) Build. 410 Bolling AFB DC 20332-6448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102 F; PROJECT NO. 2302; TASK NO. B2; WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification) "A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF NOTCHED COMPOSITE LAMINATES" (U)

12. PERSONAL AUTHOR(S)
A.S.D. Wang, E.S. Reddy, W. Binienda and Yu Zhong

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Report	FROM 30/9/84 TO 31/12/87	25 February 1988	260

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Graphite-Epoxy laminates, notches, holes, stress concentration, matrix cracks, mixed-modes, crack growth, finite element fracture mechanics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This final report contains the results of an investigation on matrix-related damage mechanisms in notched composite laminates. The theoretical approach taken follows the principles of micromechanics and the mechanics of brittle fracture at the descriptive level considered valid for the so-called ply-elasticity. Namely, the laminate is basically treated as a 3-dimensional eliastic solid which is made of distinctly anisotropi layers. Brittle fracture can initiate and propagate within any later having a weaker axis of material anisotropy, and within any one of the weaker layer interfaces due to the 3-dimensional interlaminar stresses. Owing to the particular microstructure of the laminate, growth of such sublaminate cracks constitutes a load- or time-dependent evolutionary process.

A computer simulation methodology is developed to describe the modes and the extent of damage caused initially by the presence of the notch, and subsequently by the damages themselves. Experiment using a graphite-epoxy laminate is then conducted to validate the simulation results.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☒ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr George K. Haritos	22b. TELEPHONE (Include Area Code) (202) 767-0463; 22c. OFFICE SYMBOL NA

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

FOREWORD

This is the final report for a comprehensive study on damage tolerance properties of notched composite laminates under the Air Force Grant AFOSR-84-0334. The grant was awarded to Dr. A. S. D. Wang of Drexel university with the initial grant period covering from 30 September 1984 to 31 December 1986. However, during the period from 1 September 1986 to 31 August 1987, Dr. worked at the AFOSR as visiting scientist under the Intergovernment Personnel Loan Program; Dr. C. W. Lau then served as an interim principal investigator, with the termination date of the grant extended to 31 December 1987.

The research was performed by Dr. A. S. D. Wang and his assistants: Dr. E. S. Reddy, Drexel University post-doctoral fellow, Dr. W. Binienda and Mr. Y. Zhong, Drexel University graduate students.

Major David A. Glasgow and Lt. Col. George K. Haritcs of AFOSR served successively as technical monitors during the course of this research.

INTRODUCTION

Objectives of Research.

The main objective of this research is to investigate matrix-related damage mechanisms in composite laminates that have a through-the-thickness line-notch or a small hole. A computer simulation methodology is then developed to describe the modes and the extent of damage growth caused initially by the presence of the notch (or hole), and subsequently by the damages themselves.

Theoretical Approach.

The theoretical approach taken in this endeavor followed the principles of micromechanics and the mechanics of brittle fracture at the descriptive level considered valid for the so-called ply-elasticity. Namely, the laminate is basically treated as a 3-dimensional elastic solid which is made of distinctly anisotropic layers. While each layer is assumed homogeneous and endowed with a set of effective elastic constants (see, e.g. [1]), brittle fracture can initiate and propagate within any layer having a weaker axis of material anisotropy, and within any one of the weaker layer interfaces due to the 3-dimensional interlaminar stresses.

Since the propagation modes and the growth behaviors between fracture in a layer and fracture in a layer-to-layer interface differ fundamentally owing to the particular microstructure of the laminate, growth of damages in the form of sublaminate cracks constitute a load-time dependent evolutionary process. The general premise of ply-elasticity and theory of brittle fracture on which a simulation model is based has recently been discussed in detail by Wang [2].

Crack Growth Simulation.

With laminates having a through-the-thickness line-notch or a small hole, stress

concentrations and hence sublaminate damages near the notch (or hole) are expected when the laminate is loaded by externally applied load. In order to simulate the damage initiation and damage growth as a function of the applied load, a 3-dimensional analysis of the stress field near the notch or hole must be first performed. Such a stress field, however, contains regions of stress concentration caused not only by the notch (or hole) itself in the usual sense, but also by the interaction of the free edges of the notch (or hole) with the layer interfaces known as free-edge effect [3].

In addition, if one or more sublaminate cracks have already initiated near the notch (or hole), the stress field disturbed by the presence of these cracks and the new conditions for these cracks to grow must be continuously analyzed.

Clearly, to effectively analyze such a complex system requires, as a prerequisite, an efficient and accurate finite element computational routine on one hand, and a set of physically consistent material conditions that govern the various crack growth behaviors on the other. Of course, the finite element routine must be developed in accordance within the basic confines of ply-elasticity and the theory of fracture mechanics. Similarly, material conditions governing the various crack growth behaviors must be determined independent of the laminate geometry, both in its overall shape and its lamination structure.

Finally, the simulation methodology must be validated by experiment in which actual growth of sublaminate damages is recorded as a function of the applied load. The recorded damage must be measured in quantity units consistent with those simulated numerically so that a direct comparison between the two can be made.

Major Tasks Performed.

Within the context of the forgoing discussions, the following major tasks have been performed during the course of the research:

1. Development of a 3-dimensional finite element code based on ply-elasticity and the linear theory of fracture mechanics. The code is capable of simulating the initiation and growth mechanisms of sublaminate cracks

expected to occur in certain notched laminates when they are specifically loaded.

2. Development of rigorous solutions based on anisotropic elasticity and fracture mechanics for a crack problem similar to that anticipated to occur in laminates but mathematically tractable without compromising accuracy. The same problem is then analyzed by means of the developed finite element code. Comparison of results from the two independent solution methods adjudicates the general accuracy of the finite element method.

3. Experiment to establish material conditions that govern the initiation and growth behaviors of the kinds of cracking anticipated to occur in notched laminates. The is accomplished by testing a family of specially designed specimens in which the anticipated cracking occurred, and by simulations of the observed cracking using the developed finite element code.

4. Validation of the simulation method by testing actual laminates that have through-the-thickness line-notches or small holes. Comparisons are then made between the test results and the simulation results, which display the adequacy and/or limitations of the simulation methodology.

In the next section, specifics in each of the tasks are discussed in more detail along with highlights of the results obtained therein. The actual results and the manner in which these results are obtained have been reported in open literature. Four full-length papers and one computer code with user's guide are appended to this report for reference.

The last section outlines a set of concluding remarks pertinent to the major themes of this research.

SPECIFIC TASKS AND RESULTS

3-D Finite Element Code.

As mentioned, the finite element code is developed on the basis of ply-elasticity and the theory of linear fracture mechanics. Its main functions are

1. To compute the 3-D stress field in a laminate of given lamination structure, overall laminate shape, manner of loading, the exact geometry and location of the notch. Because of the expected stress concentrations near the notch region, the code is capable of generating the desired mesh in the region around the notch. The computed stress field provides 6 stress components at any point. In general, stress distribution on any specified plane can be displayed graphically in various isometric forms.

2. To compute the strain energy release rates at any crack-tip with specified direction of propagation. If one or more cracks are already present near the notch, the code can compute the associate stress field as well as the strain energy release rate at one of the crack tip. In cases where the crack may propagate in mixed-modes, then the energy release rate corresponding to each mode can also be calculated. The calculated strain energy release rates are expressed in terms of the appropriately unit for the applied load.

Input data required to run the code include the geometry for the overall laminate specimen shape, the applied load and boundary conditions, the laminate stacking sequence and fiber orientations, the effective elastic constants (including thermal expansion coefficients if appropriate) for each of the laminating layers relative to their respective principal material axes, the location, size and orientation of the notch, and the suspected matrix crack or delamination near the notch.

Appendix V contains the user's guide in which a considerable detail about the code

is discussed. To help run the code, illustrative examples are provided with explanations and actual input /output results. A list of the source code, written in Fortran-IV, is also included.

Assessing the Accuracy of the Finite Element Method.

As the developed finite element code is to be used to compute both the stress fields and the fracture quantities for small cracks in layered, anisotropic solids, an effort is made to assess the numerical accuracy the code can provide. To this end, a problem of an ideal overall configuration and loading condition is treated rigorously on the basis of the anisotropic theory of elasticity and fracture mechanics.

The specific problem treated is a unidirectional laminate of infinite domain as illustrated in Figure 1. The laminate contains initially a kink crack and is loaded in uniform tension applied off-axis, making an arbitrary angle θ with the fibers. The base of the kink crack is normal to the applied tension while the kink itself is in the fiber direction. Thus, the problem is one that involves self-similar, mixed-mode fracture at the kink tip. Within the frame work of elasticity theory and linear fracture mechanics, the problem can be formulated exactly and solved rigorously by means of singular integrals and the boundary collocation method.

Solutions to this rigorously formulated problem serve as branch mark from which the finite element solutions can be compared. As it turns out, it is possible to tune the finite element shape and mesh selections in order to yield as accurate numerical results as the rigorous solutions.

Detailed development of this effort has been published in the paper entitled "Fracture due to A Kink Crack in Unidirectional Fiber reinforced Composites." This paper is appended here as Appendix I.

Establishing A Mixed-Mode Fracture Criterion.

Another essential element in the present effort to simulate mixed-mode sublaminate crack is to ascertain the material condition under which the crack propagates. The problem

is complicated by the fact that fracture of different modes often involves different mechanisms at the microscale, which in turn result in different material conditions for propagation. For fracture propagating in arbitrary combination of modes, a general set of conditions is need. This, however, is not always possible without actually specifying the material.

In the present work, the AS4-3501-06 graphite-epoxy composite system is used in all experiments and simulations. To establish the desired mixed-mode fracture criterion for matrix cracks in this material, a test specimen is designed which can yield crack propagation under 28 different mixed-mode conditions. The test specimen is shown in Figure 2.

It is an off-axis unidirectional tensile coupon with a pair of side notches cut normal to the applied tension. At the critical loading, a kink crack is initiated at one the notch tips and is propagated along the fiber direction in mixed-mode. By varying the off-axis angle θ and the notch depth, the nature of the mode-mix as well as the critical conditions can thus be altered.

Correlation between experiment and finite element analysis concludes that a useful criterion governing mixed-mode fracture in this material appears to be the total strain energy release rate that exists at the crack tip.

The details of this subject have been included in the paper entitled "A Criterion for Mixed-Mode Matrix Cracking in Graphite Epoxy Composites." This paper is appended here in Appendix II.

Simulation of Matrix Cracks in Notched Laminates.

For simulation of matrix crack growth in laminates, the graphite-epoxy (AS4-3501-06) $[0_2/90_2]_S$ laminate coupon is chosen. The dimension of the actual coupon is 1" wide and 9" long; it is notched in two different forms: (a) a pair of side notches and (b) a small center hole. The applied load is uniaxial tension. Under the applied loading, both in-ply matrix cracks and interply delaminations are expected to occur and grow with the increasing load. In particular, these cracks can occur interactively. It should also be emphasized that in all cases the resulting sublaminate cracks propagate in mixed-modes of various degree of mode-mix.

Evolution of the matrix cracks and delamination in the specimen is both recorded in

experiment and simulated independently by the finite element routine.

Results from this part of the study have been reported in two papers entitled "Three-Dimensional Simulation of Crack Growth in Notched Laminates," and "Simulation of Matrix Cracks in Composite Laminates Containing A Small Hole." These papers are appended here as Appendix II and Appendix IV, respectively.

CONCLUSIONS

In this research program, a simulation method is developed to describe the evolution of matrix cracks in the vicinity of notches in composite laminates. The method is based on a generic approach of the problem in which actual cracking mechanisms are closely modeled. Still, these mechanisms are extremely complex and the simulation has to resort to some degree of idealization. This then causes discrepancies between the simulation and experiment, as is evident by the results reported in the papers appended herein. It is conceivable that these difficulties could be considerable removed if more is known about the interactive mechanisms of the various cracks at the microscopic scale and if a more realistic simulation technique becomes available.

REFERENCE

[1] Tsai, S. W. and Hahn, H. T., "Introduction to Composite Materials," Technomic Pub., Lancaster, Pa. 1980.

[2] Wang, A. S. D., "Fracture Mechanics of Sublaminate Cracks in Composite Materials," Composite Technology Review, Vol. 6, 1984, pp. 45-62.

[3] Wang, A. S. D. and Crossman, F. W., "Some New Results on Edge Effects in Symmetric Composite Laminates," Journal of Composite Materials, Vol. 11, 1977, pp. 92-108.

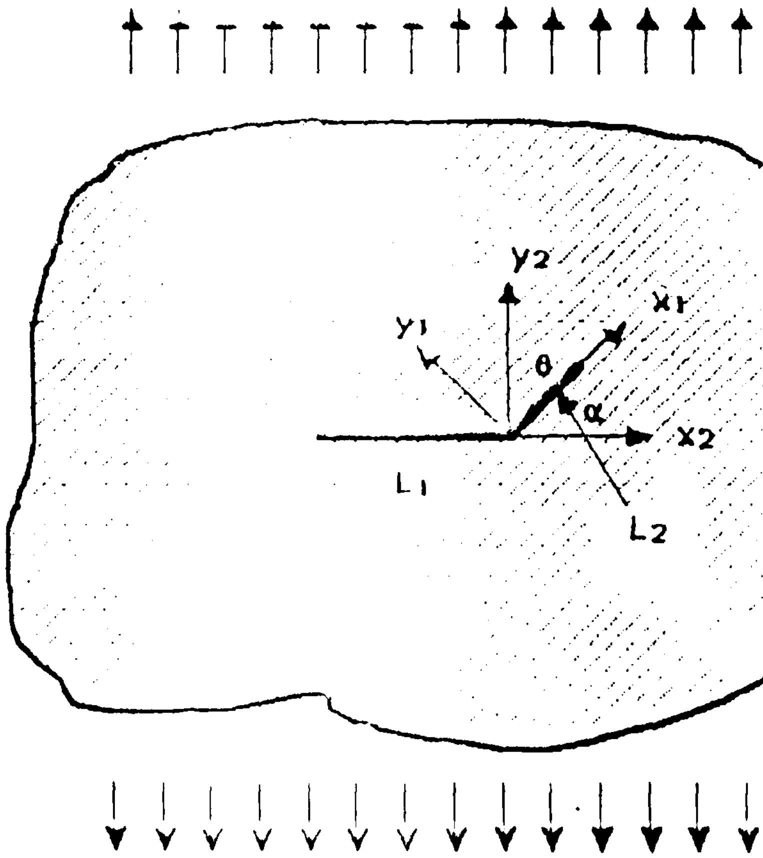


Figure 1. Kink crack in an infinite unidirectional laminate subjected to uniform tension.

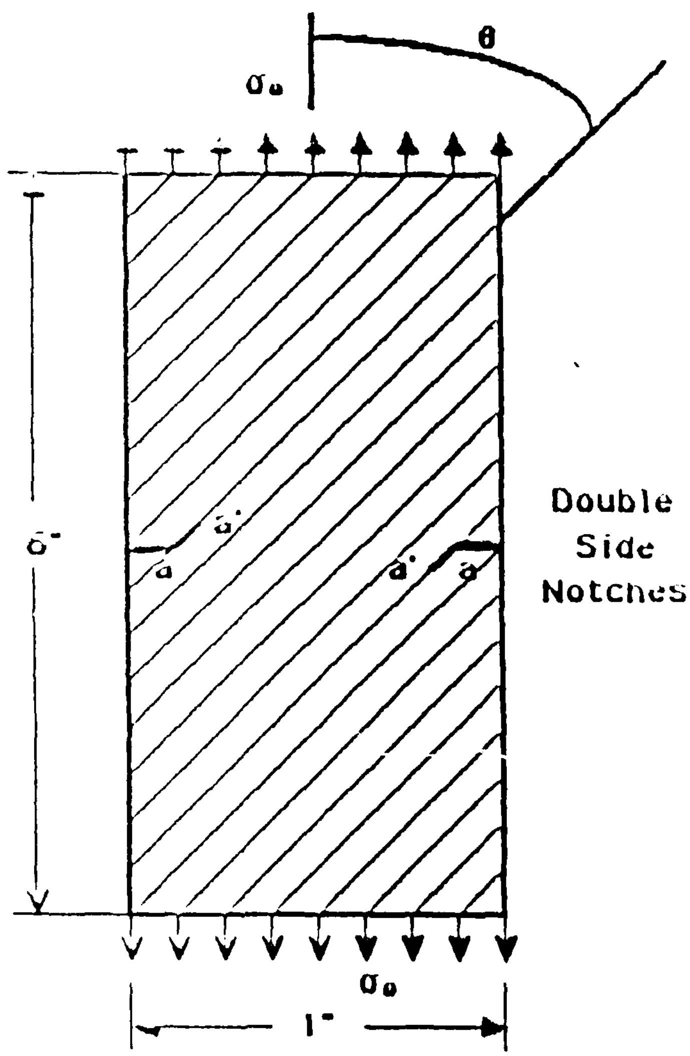


Figure 2. Geometry of test specimen used to establish mixed-mode fracture criterion.

A COMPREHENSIVE STUDY

ON DAMAGE TOLERANCE PROPERTIES OF

NOTCHED COMPOSITE LAMINATES

Appendix I

Fracture due to A Kinked Crack

in Unidirectional Fiber Reinforced Composites"

Paper presented at the ASME Winter Annual Meeting, Boston, 1987;
also in Damage Mechanics in Composites, AD-12, ASME, 1987. pp. 73-81.

FRACTURE DUE TO A KINKED CRACK IN UNIDIRECTIONAL FIBER REINFORCED COMPOSITES

W. Binienda and A. S. D. Wang
Department of Mechanical Engineering and Mechanics
Drexel University
Philadelphia, Pennsylvania

F. Delale
Department of Mechanical Engineering
The City College of New York
New York, New York

ABSTRACT

This paper presents an analysis for a kinked crack in a unidirectionally fiber reinforced composite plate. The plate is assumed infinite and contains a through-thickness crack of initial length L_1, which makes an angle θ with the direction of fibers. When the plate is subjected to a far-field uniform tensile stress normal to the crack, the crack can only propagate in the preferential direction of fibers due to the weak strength of the fiber-matrix interface. The result is a kinked crack propagating in mixed mode, with the degree of modal mixture depending on the angle θ and the ratio between the length of the kink L_2 and the length of the initial crack L_1.

To determine the parameters relevant to mixed-mode fracture at the tips of the kinked crack, the problem is formulated in terms of singular integral equations with generalized Cauchy kernels. The resulting system of equations is then solved numerically employing a Gaussian quadrature and the collocation method. Stress intensity factors, k_1 and k_2, and the strain energy release rates, G_I and G_{II}, of the kinked crack are obtained for various values of θ and L_2/L_1 ratios.

1. INTRODUCTION

Failure in fiber reinforced polymeric composites frequently occurs in the form of matrix cracks due to weak fiber/matrix interface strength. Depending on the local fiber geometry, a matrix crack may propagate in the preferential fiber direction under mixed-mode conditions. Invariably, the relevant fracture parameters which govern matrix crack propagation are dominated by the anisotropic properties of the material. This makes it necessary to formulate an anisotropic criterion for fracture propagation.

Within the frame work of the original Griffith theory for brittle fracture, a number of mixed-mode crack propagation criteria have been used for various types of materials, including fiber reinforced composites [1-6]. Sih [5,6], for example, proposed a criterion based on the local strain energy density. Others have used critera in the general form of $f(k_I,k_{II})=k_{eff}$. In the experiment by Wu [3], who tested notched balsa wood and unidirectional fiber glass reinforced composite plates, the fracture criterion $(k_I/k_{Ic})+(k_{II}/k_{IIc})^2=1$ was shown to apply.

In a series of recent papers by Wang, Crossman, et. al. [7-10], the critical energy release rate G_{Ic} was used as a criterion for the initiation and propagation of mode-I cracks in multi-layered laminates. When the crack is blunted by a local fiber or layer interface, the crack would kink and a mixed-mode or shear-dominated fracture would result. In this case, the total critical energy release rate $(G_T)_c$ has been employed as a criterion [11].

Regardless of the form of the fracture criteria, it is essential to treat the crack conditions correctly and determine the associated fracture parameters accurately.

Fracture problems in homogeneous anisotropic materials have been rigorously studied, see e.g. [12-15]. But for fracture in fibrous composites, material inhomogeneity and the associated microstructure often prevent an analytical solution. A numerical technique such as the finite element method is employed, without a rigorous interogation of the fracture conditions near the crack tip.

This paper treats a kinked crack in a unidirectionally fiber reinforced composite plate. The plate is assumed to contain a through-thickness crack of initial length L_1, which makes an angle θ with the direction of fibers. When the plate is subjected to a uniform far-field tensile stress normal to the crack, the crack can only propagate in the direction of the fiber because of the weak strength of the fiber-matrix interface. Thus, a kinked crack is induced propagating in mixed mode. Clearly, the nature of the propagation depends on the kink angle θ, the lengths of the kink L_2 and the main crack L_1.

To determine the parameters relevant to mixed-

mode fracture at the tips of the kinked crack, first the problem of two separate cracks embedded in an infinite orthotropic plate is considered. Namely, one crack is the main crack of length L_1 and the second crack of length L_2 is assumed to lie along the direction of fibers. The line of L_2 intersects the line of L_1 at the origin of the x-y coordinates as shown in Figure 1. Using the crack surface derivatives as unknown, the problem is formulated on the basis of two-dimensional theory of elasticity and the field equations are expressed in terms of singular integrals with Cauchy type kernels. The system of integral equations is then solved numerically by employing a Gaussian quadrature and the collocation method.

Next, the actual kinked crack is considered. This is accomplished by letting the approaching tips of the kink and the main cracks to touch each other at the intersect of the two crack lines. In this configuration, the singular integral equations are still valid but some of the kernels become singular, giving rise to generalized Cauchy kernels. In fact, it is shown that at the point of touch the stresses are singular and the power of singularity is different from 1/2. Thus, for the kinked crack geometry, a set of singular integral equations with singular kernels is solved. Stress intensity factors, k_1 and k_2, and strain energy release rate components G_I and G_{II}, at the tips of the kinked crack are obtained for various values of θ and L_2/L_1 ratios. Note that the problem of the plate containing only the main crack corresponds to $L_2 \to 0$.

2. FORMULATION OF THE PROBLEM

As stated previously, the problem at hand is a kinked crack in an infinite plate, and it is treated first by considering two separate cracks as depicted in Figure 1. Let the plate be orthotropic with principal directions x_1 and y_1. The far-field uniform tension is applied in the direction of y_2 which makes angle θ with y_1. The main crack of length L1 lies on the x_2 axis, while the inclined crack of length L_2 (the future kink) lies on the x_1 axis (which is the direction of the fibers). The stress fields for the invidual cracks are first solved, and the stress field for the interacting cracks is then obtained by superposition. A brief outline of the solution procedures is given below; details are contained in Reference [16].

Crack Parallel to the Fibers.

For the crack parallel to the fibers, the governing field equation is expressed in terms of the stress function $F_1(x_1, y_1)$ in the principal coordinates (x_1, y_1):

$$\frac{\partial^4 F_1}{\partial x_1^4} + \beta_2 \frac{\partial^4 F_1}{\partial x_1^2 \partial y_1^2} + \beta_1 \frac{\partial^4 F_1}{\partial y_1^4} = 0 \qquad (1)$$

where

$$\beta_1 = \frac{a_{11}}{a_{22}} ; \qquad \beta_2 = \frac{2a_{12} + a_{66}}{a_{22}} \qquad (2)$$

and

$$a_{11} = \frac{1}{E_{LL}} ; \quad a_{12} = -\frac{\nu_{LT}}{E_{LL}} ; \quad a_{22} = \frac{1}{E_{TT}} ; \quad a_{66} = \frac{1}{G_{LT}} \qquad (3)$$

E_{LL}, E_{TT}, G_{LT}, ν_{LT} being the engineering elastic constants for the orthotropic material.

Fourier transformation of the stress function $F_1(x_1, y_1)$ can be defined as:

$$F_1(x_1,y_1) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \Phi_1(s,y_1)\, e^{-isx_1} ds \qquad (4)$$

Substituting equation (4) in (1), Ordinary Differential Equation (ODE) with constant coeficients is obtained. The solution of such equation can be expressed as:

$$\Phi_1(s,y_1) \sim e^{\omega s y_1} \qquad (5)$$

so the following characteristic equation is obtained:

$$\beta_1 \omega^4 - \beta_2 \omega^2 + 1 = 0 \qquad (6)$$

The roots of equation (6) are: ω_1, $-\omega_1$, ω_2, $-\omega_2$, such that $Re(\omega_1) > 0$ and $Re(\omega_2) > 0$.

Taking into consideration the fact that the stress and displacements must vanish at infinity, the stress function may then be written as:

$$F_1(x_1,y_1^+) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} [A\, e^{-\omega_1|s|y_1} + B\, e^{-\omega_2|s|y_1}]\, e^{-isx_1} ds , \quad y_1 > 0$$
$$F_1(x_1,y_1^-) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} [C\, e^{\omega_1|s|y_1} + D\, e^{\omega_2|s|y_1}]\, e^{-isx_1} ds , \quad y_1 < 0 \qquad (7)$$

Using the continuity of stress at $y_1=0$ and introducing the following crack surface displacement derivatives as the new unknowns,

$$f_1(x_1) = \frac{\partial}{\partial x_1}[u(x_1,0^+) - u(x_1,0^-)] \qquad (8)$$
$$x_c < x_1 < x_d$$
$$f_2(x_1) = \frac{\partial}{\partial x_1}[v(x_1,0^+) - v(x_1,0^-)] \qquad (9)$$

the stresses may be expressed as:

$$\sigma_{x_1x_1} = \frac{1}{2\pi(\omega_1^2 - \omega_2^2)\, a_{11}} \int_{x_c}^{x_d} \left\{ \frac{f_1(t_1)\,\omega_1^3 y_1 + f_2(t_1)(t_1 - x_1)\,\omega_1}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1)\,\omega_2^3 y_1 + f_2(t_1)(t_1 - x_1)\,\omega_2}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1 \qquad (10)$$

$$\sigma_{y_1y_1} = \frac{-1}{2\pi(\omega_1^2 - \omega_2^2)\, a_{11}} \int_{x_c}^{x_d} \left\{ \frac{f_1(t_1)\,\omega_1 y_1 + f_2(t_1)\frac{t_1 - x_1}{\omega_1}}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1)\,\omega_2 y_1 + f_2(t_1)\frac{t_1 - x_1}{\omega_2}}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1 \qquad (11)$$

$$\tau_{x_1y_1} = \frac{1}{2\pi(\omega_1^2-\omega_2^2)a_{11}} \int_{x_a}^{x_b} \left\{ \frac{f_1(t_1)\omega_1(t_1-x_1) - f_2(t_1)\omega_1 y_1}{\omega_1^2 y_1^2 + (t_1-x_1)^2} \right.$$

$$\left. - \frac{f_1(t_1)\omega_2(t_1-x_1) - f_2(t_1)\omega_2 y_1}{\omega_1^2 y_1^2 + (t_1-x_1)^2} \right\} dt_1 \qquad (12)$$

For more details about the formulation one may refer to [16].

Crack Making an Angle θ with the Fibers.

In this configuration the crack is assumed to lie along the x_2 axis. For a formulation using the x_2-y_2 coordinate system the material has to be taken as fully anisotropic, giving the following governing equation in terms of the Airy Stress Function $F_2(x_2,y_2)$:

$$\frac{\partial^4 F_2}{\partial x_2^4} + \gamma_1 \frac{\partial^4 F_2}{\partial x_2^3 \partial y_2} + \gamma_2 \frac{\partial^4 F_2}{\partial x_2^2 \partial y_2^2} + \gamma_3 \frac{\partial^4 F_2}{\partial x_2 \partial y_2^3} + \gamma_4 \frac{\partial^4 F_2}{\partial y_2^4} = 0 \qquad (13)$$

where

$$\gamma_1 = -\frac{2b_{26}}{b_{22}}; \quad \gamma_2 = \frac{2b_{12}+b_{66}}{b_{22}};$$
$$\gamma_3 = -\frac{2b_{16}}{b_{22}}; \quad \gamma_4 = \frac{b_{11}}{b_{22}}; \qquad (14)$$

and

$$b_{11} = a_{11}\cos^4\theta + (2a_{12}+a_{66})\sin^2\theta\cos^2\theta + a_{22}\sin^4\theta$$
$$b_{22} = a_{11}\sin^4\theta + (2a_{12}+a_{66})\sin^2\theta\cos^2\theta + a_{22}\cos^4\theta$$
$$b_{12} = a_{12} + (a_{11}+a_{22}-2a_{12}-a_{66})\sin^2\theta\cos^2\theta$$
$$b_{66} = a_{66} + (a_{11}+a_{22}-2a_{12}-a_{66})\sin^2\theta\cos^2\theta \qquad (15)$$
$$b_{16} = [a_{22}\sin^2\theta - a_{11}\cos^2\theta + \frac{1}{2}(2a_{12}+a_{66})\cos 2\theta]\sin 2\theta$$
$$b_{26} = [a_{22}\cos^2\theta - a_{11}\sin^2\theta - \frac{1}{2}(2a_{12}+a_{66})\cos 2\theta]\sin 2\theta$$

Again following the same procedure, the stress can be expressed in terms of the crack displacement derivatives $f_3(t_2)$ and $f_4(t_2)$ as follows:

$$\sigma_{x_2x_2} = \frac{1}{2\pi}\int_{x_a}^{x_b} \left[\frac{R_1 f_3(t_2) - R_2 f_4(t_2)}{y_2(a+ib) + i(t_2-x_2)} + \frac{R_3 f_4(t_2) - R_4 f_3(t_2)}{y_2(c+id) + i(t_2-x_2)} \right.$$
$$\left. + \frac{R_5 f_3(t_2) - R_6 f_4(t_2)}{y_2(a-ib) - i(t_2-x_2)} + \frac{R_7 f_4(t_2) - R_8 f_3(t_2)}{y_2(c-id) - i(t_2-x_2)} \right] dt_2 \qquad (16)$$

$$\sigma_{y_2y_2} = \frac{1}{2\pi}\int_{x_a}^{x_b} \left[\frac{R_9 f_3(t_2) - R_{10} f_4(t_2)}{y_2(a+ib) + i(t_2-x_2)} + \frac{R_{11} f_4(t_2) - R_{12} f_3(t_2)}{y_2(c+id) + i(t_2-x_2)} \right.$$
$$\left. + \frac{R_{13} f_3(t_2) - R_{14} f_4(t_2)}{y_2(a-ib) - i(t_2-x_2)} + \frac{R_{15} f_4(t_2) - R_{16} f_3(t_2)}{y_2(c-id) - i(t_2-x_2)} \right] dt_2 \qquad (17)$$

$$\tau_{x_2y_2} = \frac{1}{2\pi}\int_{x_a}^{x_b} \left[\frac{R_{17} f_3(t_2) - R_{18} f_4(t_2)}{y_2(a+ib) + i(t_2-x_2)} + \frac{R_{19} f_4(t_2) - R_{20} f_3(t_2)}{y_2(c+id) + i(t_2-x_2)} \right.$$
$$\left. + \frac{R_{21} f_3(t_2) - R_{22} f_4(t_2)}{y_2(a-ib) - i(t_2-x_2)} + \frac{R_{23} f_4(t_2) - R_{24} f_3(t_2)}{y_2(c-id) - i(t_2-x_2)} \right] dt_2 \qquad (18)$$

where

$$f_3(x_2) = \frac{\partial}{\partial x_2}[u(x_2,0^+) - u(x_2,0^-)] \qquad (19)$$
$$x_a < x_2 < x_b$$
$$f_4(x_2) = \frac{\partial}{\partial x_2}[v(x_2,0^+) - v(x_2,0^-)] \qquad (20)$$

and R_i $i=1,2,\ldots 24$ are given in [16]. Here it must be noted that the formulation leading to expressions (16)-(18) is quite lengthy and tedious. The intermediate steps can be found in [16].

The Integral Equations.

Stress field for two-crack system is generated by superimposing the two solutions briefly described in two previous sections. It is noted that the stresses are given in different coordinate systems. Therefore the following coordinate transformations are used:

$$x_2 = x_1\cos\theta - y_1\sin\theta$$
$$y_2 = x_1\sin\theta + y_1\cos\theta \qquad (21)$$

or

$$x_1 = x_2\cos\theta + y_2\sin\theta$$
$$y_1 = -x_2\sin\theta + y_2\cos\theta \qquad (22)$$

The total solution for the stress field can be expressed in either (x_1,y_1) or (x_2,y_2). Let superscript (T) be used to denote the total stresses in either system. To satisfy the boundary conditions along $y_2=0$ and $y_1=0$ we may write:

$$\sigma^T_{y_2y_2} = -\sigma_0$$
$$x_a < x_2 < x_b \qquad (23)$$
$$\tau^T_{x_2y_2} = 0$$

and

$$\sigma^T_{y_1y_1} = -\sigma_0 \cos^2\theta \qquad x_c < x_1 < x_d \qquad (24)$$

$$\tau^T_{x_1y_1} = -\sigma_0 \sin\theta \cos\theta$$

By means of a normalization procedure by substituting the following:

$$t_2 = \frac{\tau_2(x_b - x_a)}{2} + \frac{x_b + x_a}{2}$$

$$x_2 = \frac{s_2(x_b - x_a)}{2} + \frac{x_b + x_a}{2} \qquad -1 < \tau_2, s_2 < 1 \qquad (25)$$

$$dt_2 = \frac{x_b - x_a}{2} d\tau_2$$

and

$$t_1 = \frac{\tau_1(x_d - x_c)}{2} + \frac{x_d + x_c}{2}$$

$$x_1 = \frac{s_1(x_d - x_c)}{2} + \frac{x_d + x_c}{2} \qquad -1 < \tau_1, s_1 < 1 \qquad (26)$$

$$dt_1 = \frac{x_d - x_c}{2} d\tau_1$$

(23) and (24) lead to the following system of Cauchy type singular integral equations:

$$C_{12}\int_{-1}^{1} \frac{f_2(\tau_1)}{\tau_1 - s_1} d\tau_1 + \int_{-1}^{1} K_{13} f_3(\tau_2)\, d\tau_2 + \int_{-1}^{1} K_{14} f_4(\tau_2)\, d\tau_2$$

$$= -\sigma_0 \cos^2\theta \qquad (27)$$

$$C_{21}\int_{-1}^{1} \frac{f_2(\tau_1)}{\tau_1 - s_1} d\tau_1 + \int_{-1}^{1} K_{23} f_3(\tau_2)\, d\tau_2 + \int_{-1}^{1} K_{24} f_4(\tau_2)\, d\tau_2$$

$$= -\sigma_0 \sin\theta \cos\theta \qquad (28)$$

$$C_{33}\int_{-1}^{1} \frac{f_3(\tau_2)}{\tau_2 - s_2} d\tau_2 + C_{34}\int_{-1}^{1} \frac{f_4(\tau_2)}{\tau_2 - s_2} d\tau_2 + \int_{-1}^{1} K_{31} f_1(\tau_1)\, d\tau_1$$

$$+ \int_{-1}^{1} K_{32} f_2(\tau_1)\, d\tau_1 = -\sigma_0 \qquad (29)$$

$$C_{43}\int_{-1}^{1} \frac{f_3(\tau_2)}{\tau_2 - s_2} d\tau_2 + C_{44}\int_{-1}^{1} \frac{f_4(\tau_2)}{\tau_2 - s_2} d\tau_2 + \int_{-1}^{1} K_{41} f_1(\tau_1)\, d\tau_1$$

$$+ \int_{-1}^{1} K_{42} f_2(\tau_1)\, d\tau_1 = 0 \qquad (30)$$

These equations must be solved with the following single-valuedness conditions which complete the formulation of the problem:

$$\int_{-1}^{1} f_1(\tau_1)\, d\tau_1 = 0 \qquad (31)$$

$$\int_{-1}^{1} f_3(\tau_2)\, d\tau_2 = 0 \qquad (32)$$

$$\int_{-1}^{1} f_2(\tau_1)\, d\tau_1 = 0 \qquad (33)$$

$$\int_{-1}^{1} f_4(\tau_2)\, d\tau_2 = 0 \qquad (34)$$

The expressions for the kernels K_{ij} are functions of material constants and crack geometry [16].
The system of integral equations (27-34) can be solved by using one of the Gaussian quadrature technique [17],[18]. It should be noted that this system of integral equations contain Cauchy type kernels, so the stress and strains will have a *square-root singularity and one may therefore use* the classical definition of stress intensity factors to evaluate them at the crack tips [12-14].

<u>Solution for the Kinked Cracked.</u>

The geometry of interest is that of a kinked cracked. We can arrive at that configuration by letting $x_b=0$ and $x_c=0$. In this case the integral equations (27-30) remain valid but some of the kernels become singular while approaching the tips, giving rise to a singularity of unknown power β at the apex. The singularity β can be derived by requiring the displacements of common end to match what giving the following transcendental characteristic equation:

$$\begin{aligned}
&-\pi^4\cos^4\pi\beta\, C_{12}C_{21}C_{33}C_{44} - \frac{\pi^2}{4}\cos^2\pi\beta\, C_{12}C_{34}A_{23}A_{41} \\
&- \frac{\pi^2}{4}\cos^2\pi\beta\, C_{12}C_{43}A_{24}A_{31} + \frac{\pi^2}{4}\cos^2\pi\beta\, C_{12}C_{33}A_{24}A_{41} \\
&+ \frac{\pi^2}{4}\cos^2\pi\beta\, C_{12}C_{44}A_{23}A_{31} + \pi^4\cos^4\pi\beta\, C_{12}C_{2}\,\text{[illegible]}\,C_{43} \\
&+ \frac{\pi^2}{4}\cos^2\pi\beta\, C_{21}C_{44}A_{13}A_{32} + \frac{1}{16}A_{13}A_{24}A_{3} \qquad (35) \\
&- \frac{1}{16}A_{13}A_{24}A_{32}A_{41} - \frac{\pi^2}{4}\cos^2\pi\beta\, C_{21}C_{34}A_{13}A_{42} \\
&- \frac{\pi^2}{4}\cos^2\pi\beta\, C_{21}C_{43}A_{14}A_{32} - \frac{1}{16}A_{14}A_{23}A_{31}A_{42} \\
&+ \frac{1}{16}A_{14}A_{23}A_{32}A_{41} + \frac{\pi^2}{4}\cos^2\pi\beta\, C_{21}C_{33}A_{14}A_{42} = 0
\end{aligned}$$

For the details of the derivation and definition of A_{kl} and C_{mn} one may again refer to [16].

For the same reason two conditions from (31-34) are replaced by:

$$x_d \int_{-1}^{1} f_2(\tau_1)\,d\tau_1 = x_a \int_{-1}^{1} \left[f_3(\tau_2)\sin\theta - f_4(\tau_2)\cos\theta \right] d\tau_2 \qquad (36)$$

$$x_d \int_{-1}^{1} f_1(\tau_1)\,d\tau_1 = -x_a \int_{-1}^{1} \left[f_3(\tau_2)\cos\theta + f_4(\tau_2)\sin\theta \right] d\tau_2 \qquad (37)$$

The singular integral equations have generalized Cauchy kernels and may be solved by using a Gauss-Jacobi [17] or Lobatto-Jacobi quadrature technique [22]. The stress intensity factors at the crack tips can again be derived using their classical definitions.

The Strain Energy Release Rate.

From the fracture point of view, perhaps the most important physical quantity is the strain energy release rate G. Using the usual definition [26], it can be written that:

$$G = \frac{d}{da}(U - V) \qquad (38)$$

at $x=x_d$, we may write:

$$dU - dV = \frac{1}{2}\int_{x_d}^{x_d+da} \Big[\sigma_{y_1y_1}(x_1,0)[\,v(x_1-da,0^+) - v(x_1-da,0^-)]$$

$$+ \tau_{x_1y_1}(x_1,0)[\,u(x_1-da,0^+) - u(x_1-da,0^-)] \Big] dx_1 \qquad (39)$$

The expressions of normal and shear stresses can be found using definition for stress intensity factors. Thus,

$$\sigma_{y_1y_1}(x_1,0) = \frac{k_1(x_d)}{\sqrt{2(x_1-x_d)}} + \text{higher order terms} \qquad (40)$$

$$\tau_{x_1y_1}(x_1,0) = \frac{k_2(x_d)}{\sqrt{2(x_1-x_d)}} + \text{higher order terms} \qquad (41)$$

To obtain the asymptotic expressions for $[u(x_1,0^+) - u(x_1,0^-)]$ and $[v(x_1,0^+) - v(x_1,0^-)]$, we can use equations (8) and (9). Following the procedure of derivation as in [27,28], it can be shown that :

$$G_I = \frac{1}{4}\frac{k_1^2(x_d)}{C_{12}} \qquad (42)$$

$$G_{II} = \frac{1}{4}\frac{k_2^2(x_d)}{C_{21}} \qquad (43)$$

and $$G = G_I + G_{II} \qquad (44)$$

For all the details one may refer to [16].

3. RESULTS AND DISCUSSION.

The important results are those pertaining to the kinked crack case. Here for conciseness only this case is studied in details. To determine the stress intensity factors one must first obtain the singularity β by solving equation (35), so certain material properties of orthotropic plate have to be used. The singularity β for an isotropic wedge is given in [19]. Similiar results are reported for an orthotropic wedge in [20] and [21]. The numerical values of β obtained from equation (35) for the special cases of isotropic and orthotropic materials compared closely with those computed in [19-21]. Figure 2 shows the variation of the stress singularity power ($-\beta$) with the angle θ for an isotropic an orthotropic material. For the orthotropic case the material properties are listed in Table 1.

As expected for $\theta=0$ (i.e. for a half plane) there is no singularity ($\beta\geq0$) and the singularity increases with increasing wedge angle. The value of β must eventually reach the value -0.5 (the well-known square-root singularity) for a crack (i.e. when $\theta=180°$). It is interesting to note that for some orthotropic materials the stresses may not be singular even if the wedge angle is larger than 180°.

The stress intensity factors are obtained by solving equations (27-30) in conjunction with equations (36) and (37). Since the integral equations have generalized Cauchy kernels, the collocation methods described in [17] and [22,23] are used. In the results given subsequently, the stress intensity factors are normalized with respect to the uniaxial load σ_0 and the square-root of their respective half crack length. To check the accuracy of the technique the results are first compared with the solutions of special cases that exist in the literature. Table 2. shows the comparison of the mixed-mode stress intensity factors at the tips of a kinked crack embedded in an infinite isotropic plate with those found in [24-25].

As one may infer from the Table, the results compare rather well. The stress intensity factors at the tips of a kinked crack are given in Figures 3-6. Figures 3 and 4 show the variation of the normalized stress intensity factors with respect to crack length ratio L_2/L_1 whereas Figures 5 and 6 display the same results with respect to the angle θ. Results are obtained for orthotropic as well as isotropic materials.

It is seen that (Figures 3 and 4) for a fixed angle θ, normalized $k_1(d)$ and $k_2(d)$ decrease with increasing L_2/L_1, however the strain energy release rates increase with increasing L_2/L_1 (Figures 7 and 8). So there is a very small chance for crack arrest, as it is ilustrated for the 30° plate. On the other hand for varying angle θ (Figures 5 and 6), k1(d) decreases while k2(d) first increases then decreases with increasing θ. For this case the total strain energy is not a monotonous function of θ (Figures 9 and 10). Thus the resistance to fracture may strongly depend on the direction of reinforcing fibers. It may be seen that (Figure 9) for isotropic materials G is a monotonically decreasing function with increasing θ, while for the orthotropic material used in the calculations (Figure 10), G first decreases then

increases due to strong influence of mode-II component of the strain energy release rate.

REFERENCES

[1] F. Erdogan and G.C. Sih, "On the Crack Extension in Plates Under Plane Loading and Transverse Shear", Trans. ASME, J. Basic Engng., Vol.85, pp. 519-527, 1963.

[2] J.G. Williams and P.D. Ewing, "Fracture Under Complex Stress - The Angled Crack Problem", Int. J. Fracture, Vol.9, pp. 441-445, 1972.

[3] E.M. Wu, "Application of Fracture Mechanics to Anisotropic Plates", Trans. ASME, J. Appl. Mech., Vol. 34, pp. 967-974, 1967.

[4] B. Cotterell, "The Influence of the Stress Distribution at the Tip of a Crack", Int. J. Fracture, Vol. 2, pp. 526-533, 1966.

[5] G.C. Sih, "Some Basic Problems in Fracture Mechanics and New Concepts", Engng. Fracture Mech., Vol. 5, pp. 365-377, 1973.

[6] G.C. Sih, "Strain Energy Density Factor Applied to Mixed Mode Crack Problems", Int. J. Fracture, Vol. 10, pp. 305-321, 1974.

[7] A.S.D. Wang and F.W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1, An Energy Method", J.Composite Materials, Vol. 4, pp. 71-87, 1980.

[8] F.W. Crossman and W.J. Warren, A.S.D. Wang and G.E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2, Experimental Correlation", J. Composite Materials, Vol. 14, pp. 88-106, 1980.

[9] A.S.D. Wang, "Growth Mechanisms of Transverse Cracks and Ply Delamination in Composite Laminates", in Advances in Composite Materials, Proc. ICCM-III, pp. 170-182, 1980.

[10] A.S.D. Wang, "Fracture Mechanics of Sub-Laminate Cracks in Composite Laminates", Proc. AGARD Struct. Matls. Specialist Meeting, London, 1983.

[11] A.S.D. Wang, N.N. Kishore and W. Feng, "On Mixed Mode Fracture in Off-Axis Unidirectional Graphite Epoxy Composites", ICCM-4, Tokyo, Japan, 1982.

[12] F. Delale and F. Erdogan, "The Problem of Internal and Edge Cracks in an Orthotropic Strip", ASME, Journal of Applied Mechanics, Vol. 44, pp.237-242, 1977.

[13] F. Delale, I. Bakirtas and F. Erdogan, "The Problem of an Inclined Crack in an Orthotropic Strip", ASME, Journal of Applied Mechanics, Vol. 46, pp. 90-96, 1979.

[14] F. Delale and F. Erdogan, "Bonded Orthotropic Strips with Cracks", Int. J. of Fracture, Vol. 15, pp.343-364, 1979.

[15] F. Delale, "Fracture Analysis of Notched Composites", ASME, Journal of Engineering Materials and Technology, Vol. 109, pp. 27-35, 1987.

[16] A.S.D. Wang, W. Binienda, Yu Zhong and E.S. Reddy, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites", NADC Technical Report, April,1987.

[17] F. Erdogan, "Mixed Boundary-Value Problems in Mechanics", Mechanics Today, edited by S. Nemat-Nasser, Vol. 4, pp. 1-86, Pergamon Press, Oxford, 1978.

[18] P.S. Theocaris and N.I. Ioakimidis, "Numerical Integration Methods for the Solution of Singular Integral Equations", Quarterly of Appl. Mathematics, pp. 173-183, 1977.

[19] M.L. Williams, "Stress Singularities Resulting from Various Boundary Conditions in Angular Corners of Plates in Extension", J. Appl. Mech., Vol. 19, pp. 526-528, 1952.

[20] M.C. Kuo and D.B. Bogy, "Plane Solutions for the Displacement and Traction-Displacement Problems for Anisotropic Elastic Wedges", J. Appl. Mech., Vol. 41, pp. 197-202, 1974.

[21] M.C. Kuo and D.B. Bogy, "Plane Solution for Traction Problems on Orthotropic Unsymmetrical Wedges and Symmetrically Twinned Wedges", J. Appl. Mech., Vol. 41, pp. 203-208, 1974.

[22] N.I. Ioakimidis and P.S. Theocaris, "The Numerical Evaluation of a Class of Generalized Stress Intensity Factors by Use of the Lobatto-Jacobi Numerical Integration Rule", Int. J. of Fracture, Vol. 14, pp. 469-484, 1978.

[23] N.I. Ioakimidis and P.S. Theocaris, "On the Selection of Collocation Points for the Numerical Solution of Singular Integral Equations with Generalized Kernels Appearing in Elasticity Problems", Computers & Structures, Vol. 11, pp. 289-295, 1980.

[24] G.D. Gupta, "Strain Energy Release Rate for Mixed Mode Crack Problem", ASME, 76-WA/PVP-7, 1977.

[25] S.N. Chatterjee, "The Stress Field in the Neighborhood of a Branched Crack in an Infinite Elastic Sheet", Int. J. of Solids & Structures, Vol. 11, pp. 521-538, 1975.

[26] G.P. Cherepanov, "Mechanics of Brittle Fracture",

McGraw-Hill, N.Y., 1979.

[27] F. Delale, "Mode III Fracture of Bonded Nonhomogeneous Materials", Engineering Fracture Mechanics, Vol. 22, pp. 213-226, 1985.

[28] S. Krenk, "Influence of Transverse Shear on an Axial Crack in a Cylindrical Shell", Int. J. Fracture, Vol. 14, pp. 123-143, 1978.

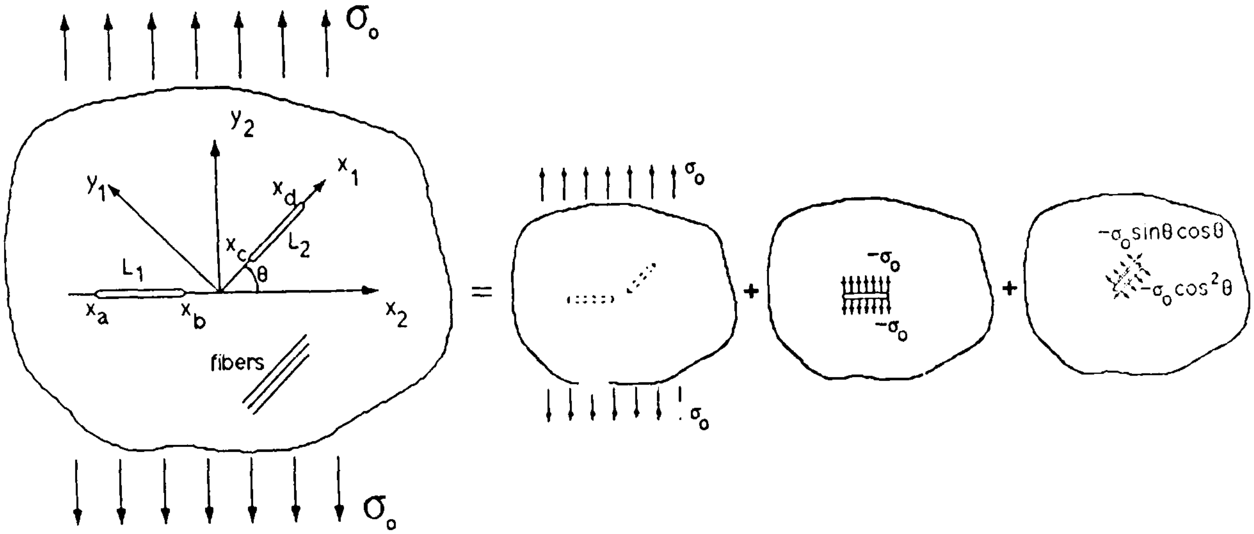


Figure 1. Superposition Scheme for the Infinite Composite Plate with Two Embedded Cracks.

Table 1. Material constants for orthotropic plate.

E_L	21.08 e+06 psi.
E_T	1.5 e+06 psi.
G_{LT}	0.98 e+06 psi.
ν_{LT}	0.3

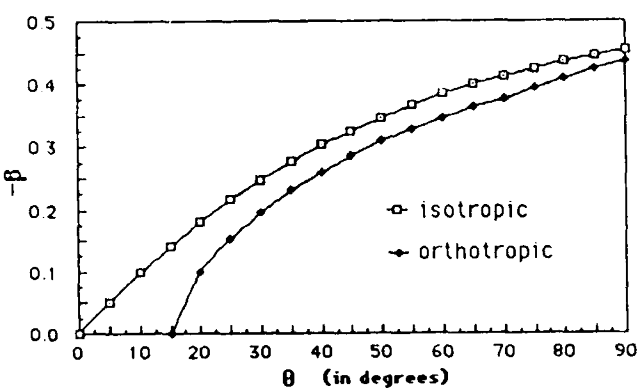


Figure 2. Variation of the Stress Singularity Power (- β) with the Angle θ.

Table 2. Comparison of present solution with references for the special case of isotropic material.

θ		k1(a)	k2(a)	k1(d)	k2(d)
30°	[24]	1.3559	0.0327	1.0873	0.6833
	[25]	1.3508	0.0325	1.0830	0.6804
	Present	1.3421	0.0328	1.0949	0.6855
45°	[24]	1.2902	0.0211	0.7463	0.8405
	[25]	1.2887	0.0208	0.7438	0.8377
	Present	1.2732	0.0217	0.7546	0.8450
60°	[24]	1.2221	-0.0109	0.3900	0.8319
	[25]	1.2194	-0.0116	0.3822	0.8292
	Present	1.2082	-0.0108	0.3941	0.8350

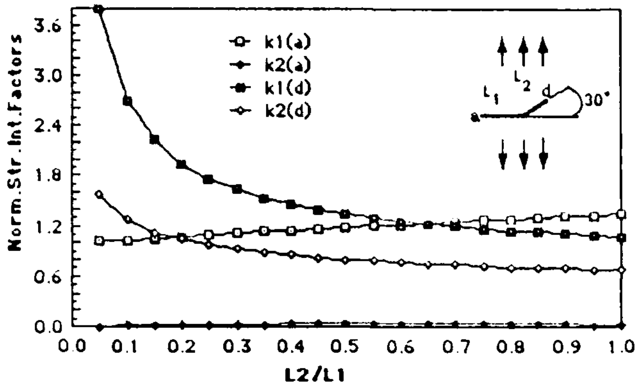


Figure 3. Variation of the Normalized Stress Intensity Factors with L_2/L_1. Isotropic Case.

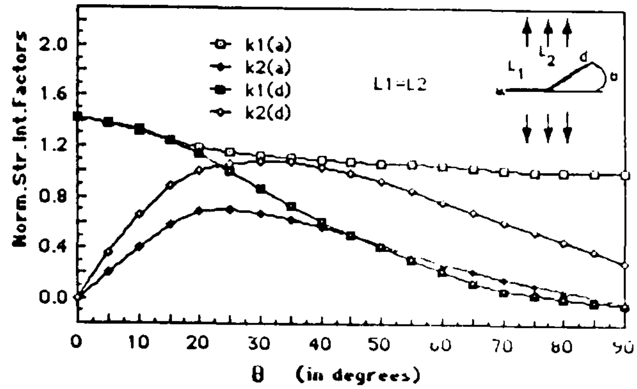


Figure 6. Variation of the Normalized Stress Intensity Factors with the angle θ. Orthotropic Case.

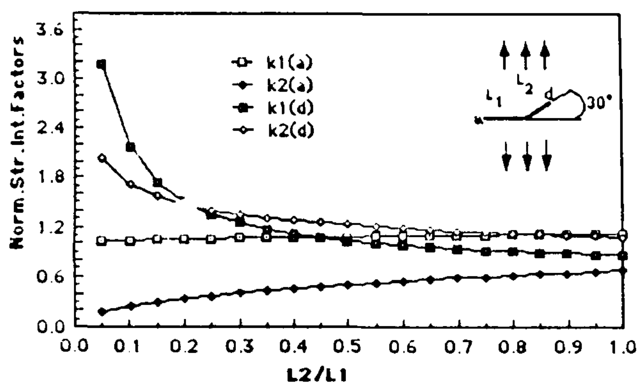


Figure 4. Variation of the Normalized Stress Intensity Factors with L_2/L_1. Orthotropic Case.

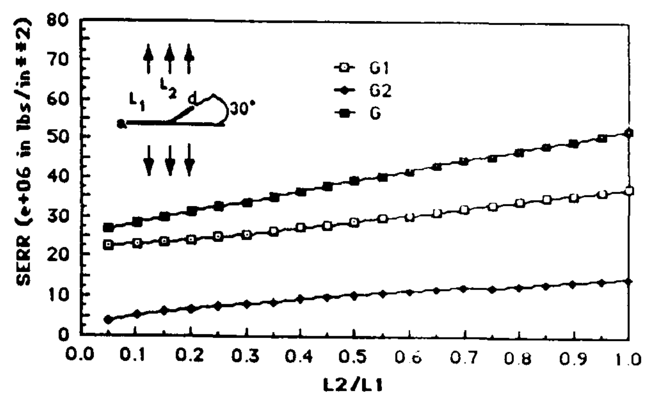


Figure 7. Variation of the Strain Energy Release Rates with L_2/L_1. Isotropic Case.

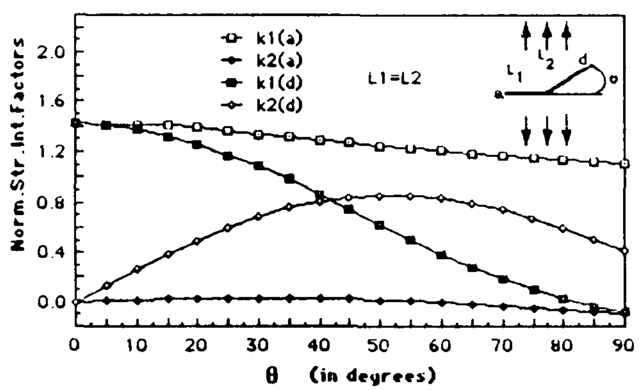


Figure 5. Variation of the Normalized Stress Intensity Factors with the angle θ. Isotropic Case.

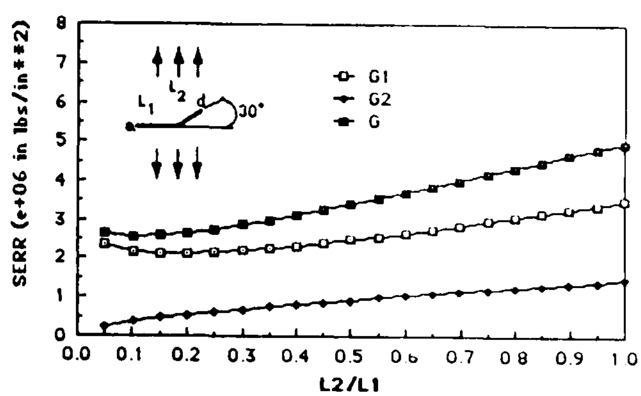


Figure 8. Variation of the Strain Energy Release Rates with L_2/L_1. Orthotropic Case.

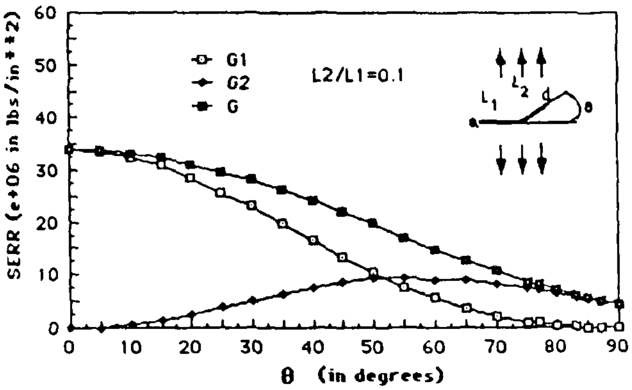


Figure 9. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle θ. Isotropic Case. (e_x=1, $L_2 \rightarrow 0$).

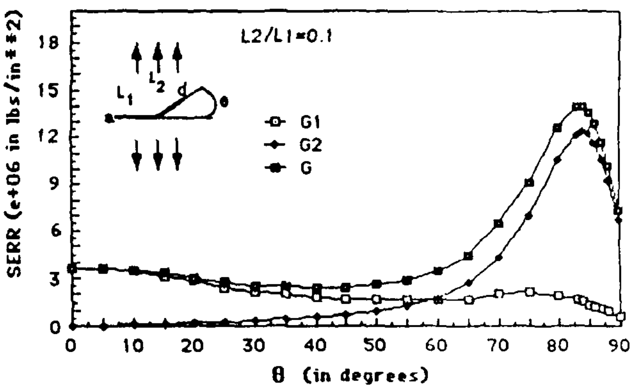


Figure 10. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle θ. Orthotropic Case. (e_x=1, $L_2 \rightarrow 0$).

A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF NOTCHED COMPOSITE LAMINATES

Appendix II

A Criterion for Mixed-Mode Matrix Cracking in Graphite-Epoxy Composites

Paper presented at the ASTM 9th Symposium on Composites, Reno, 1988;
also to appear in ASTM STP.

A CRITERION FOR MIXED-MODE MATRIX CRACKING IN GRAPHITE-EPOXY COMPOSITES

W. Binienda, A. S. D. Wang, Y. Zhong and E. S. Reddy
Department of Mechanical Engineering
Drexel University
Philadelphia, PA 19104

ABSTRACT: In this paper, mixed-mode matrix fracture in graphite-epoxy composites has been studied. Experimental investigation was conducted on a family of doubly side-notched unidirectional off-axis specimens. By varying the notch depth and the off-axis angle, a total of 28 fracture conditions of differing mixed-mode ratios was produced. Fracture analysis of the test data suggested that the total strain energy release rate is a suitable material condition for mixed-mode matrix cracking in graphite-epoxy composites.

KEYWORDS: graphite-epoxy, mixed-mode matrix fracture, strain energy release rates, finite element analysis, mixed-mode fracture criterion.

Structural composites, notably laminates made of unidirectional tape systems, can sustain extensive matrix cracking before the load carrying fibers fail. Matrix cracking usually occurs at low stress level due to weak interfacial bond strength between matrix and fiber, and between laminating plies. Thus, propagation of matrix cracks in laminates either follows the fiber-matrix interface or the ply-to-ply interface, or both.

Fig. 1 is an x-radiograph taken from a graphite-epoxy $[0_2/90_2]s$ laminate having a center-notch. When the laminate is loaded in uniaxial tension, extensive damage in the form of matrix cracks near the notch can be observed. At this phenomenological scale, matrix cracking can be classified into two major modes. Namely, the intra-ply cracking (fiber-wise splitting) which occurs inside a ply and propagate along the fibers; and the inter-ply cracking

(delamination) which occurs in the interface between two adjacent plies.

In Fig. 1, the four vertical cracks were initiated first near the hole and then propagated along the fibers in the 0^0-ply. The driving force here is the interfacial shear due to load-transfer from the fiber bundle cut by the hole to the fiber bundle which is uncut. Because of the constraint stemming from bonding between the 0^0 and the 90^0 plies, the vertical splits propagated stably with the applied tension.

As the vertical cracks propagated away from the hole, another mode of load-transfer then took place between the cracked 0^0-ply and the uncracked 90^0-ply. Secondary inter-ply stresses along the roots of the vertical cracks were then induced, which then initiated delamination in the 0/90 interface.

Fracture analysis of the cracked specimen at each major form of cracking reveals that the corresponding crack-tip stress fields are complex and the associated propagation involves both opening and shearing modes.

Model simulation for intra-ply fiber-wise matrix cracking and inter-ply delamination has recently been performed using the strain energy release rate method [1]. This method, when limited to mode-I propagation conditions, has proven useful for modeling brittle matrix cracks in graphite-epoxy systems. In such cases, it is necessary to determine the strain energy release rate G_I at the crack front as driving force, and to validate the corresponding critical strain energy release rate G_{Ic} as material resistance [1].

MIXED-MODE FRACTURE CRITERIA

As illustrated in Fig. 1, most matrix cracking in laminates involves mixed opening and shearing modes. However, the applicabilty of the energy release rate criterion to mixed-mode cracking has not been as firmly established.

Several studies aimed at establishing criteria for mixed-mode matrix

cracking in unidirectional laminates have been conducted in the past using graphite-epoxy composites. Wilkins, et. al. [2] and Ramkumar, et.al [3] used the cracked-lap shear specimen loaded in uniaxial tension to induce mixed mode-I and mode-II delamination between the lap-layer and the substrate layer. By varying the thickness of the lap-layer relative to the substrate layer, mixed-mode ratio, G_{II}/G_{I}, ranging from 0.35 to 0.45 could be obtained. They observed that the total strain energy release rate $(G_I+G_{II})_C$ obtained under mixed-mode conditions is slightly greater than G_{IC} obtained under pure mode-I conditions. Bradley and Cohen [4] used a cantiliver split-beam specimen loaded by a pair of upward and downward loads applied at the tip of the cantiliver. Variation of the mixed-mode ratio G_{II}/G_I was achieved by changing the ratio of the upward and downward loads. Mixed-mode conditions with G_{II}/G_I ratios ranging from 0 to about 0.6 were produced. They observed that, in composite systems made of brittle matrix, the measured total strain energy release rate $(G_I+G_{II})_C$ increased with G_{II}/G_I; but it decreased slightly with G_{II}/G_I in systems of ductile matrix. Wang, et. al. [5] used a double side-notched, off-axis unidirectioal laminate specimen loaded in axial tension. By varying the off-axis angle from 0^0 to 90^0 and the depth of the notches, mixed-mode conditions with G_{II}/G_I ratios ranging from 0 to about 2.5 were achieved. They found that the total strain energy release rate $(G_I+G_{II})_C$ increased with G_{II}/G_I up to about $G_{II}/G_I = 1.5$; it then remained constant for G_{II}/G_I between 1.5 and 2.5.

Russell and Street [6] used specimens of four different configurations and obtained critical strain energy release rates for a wide range of mixed-mode cracking conditions, including pure mode-II cracking. They showed

that the critical strain energy release rates depended on the test specimen and test method used; hence, a general criterion for all the mixed-mode matrix cracking cases tested could not be established.

One possible reason for the lack of a general criterion has been attributed to the manner in which fracture analysis of the test specimens was performed. In the case of a beam-like specimen, the approximate beam theory was employed, while in the case of the plate-like specimen, a finite element plate model was constructed. These analysis methods lacked the required precision to treat complicated singular stress fields, to simulate the actual loading conditions or to properly represent the exact configuration of the cracked specimens. Significant numerical errors could result in the computed fracture quantities, especially for mixed-mode cracking.

Another possible reason stems from uncertainties about the fracture mechanisms associated with pure mode-II cracking. Specifically, ideally pure mode-II cracking is difficult to simulate by tests. In actual experiment, pure mode-II propagation is often accompanied by some amount of friction between the cracked surfaces. The fracture analysis models do not include any such friction mechanisms. A separate criterion may be needed for pure mode-II cracking.

THE PRESENT INVESTIGATION

In this paper, a mixed-mode criterion is suggested for matrix cracks propagating in graphite-epoxy composites. This criterion is based on analysis of test data using specimems of varying cracked configurations, which provide mixed-mode fracture conditions with G_{II}/G_I ratios ranging uniformly from 0 to about 3. The case of predominantly mode-II ($G_{II}/G_I > 3$) or pure mode-II ($G_I = 0$) is excluded. Fracture analysis of the test specimens is performed using a finite element crack growth simulation model, as exact solutions for the test

specimen configurations cannot presently be obtained. The accuracy of the simulation model is, however, adjudicated by comparing results of problems of similar crack configurations whose solutions can also be found rigorously.

Experiment

The specimen used in the experiment is a notched off-axis tension coupon prepared from a unidirectional laminate made of Hercules AS4-3501-06 graphite-epoxy prepreg tape. Fig. 2 depicts the general configuration of the coupon. The overall dimension is 23 cm long and 2.5 cm wide. Excluding the 4 cm end-tabs, the clear section of the coupon is about 15 cm in length. The pair side-notches are introduced at the mid-section by an 8-mil (0.2 mm) thick diamond saw.

The depth of the side-notch a and the off-axis angle θ (between the applied tension and the direction of the fibers) are varied in the test program as follows:

$$\theta = 0^0, 5^0, 10^0, 15^0, 20^0, 25^0, 90^0$$

$$a = 2.5 \text{ mm}, 3.2 \text{ mm}, 3.8 \text{ mm}, 4.5 \text{ mm}$$

As depicted in Fig. 3, the coupon can initiate a kink crack (denoted as a') at the side-notch tip and propagate in the fiber direction when the applied tension σ_0 reaches some critical value. The propagation is generally mixed with modes I and II. The degree of mix is determined solely by the angle θ, if the notch depth a is held constant. Conversely, if θ is held fixed, the critical applied tension at the onset of the kink is determined by the notch depth, a.

In this experiment, a total of 28 mixed-mode fracture conditions were created by varying θ and a as mentioned. This has provided fractures with G_{II}/G_I ratios ranging uniformly from 0 to about 3. It should be noted that mixed-mode matrix fracture in such a wide G_{II}/G_I ratio range has not been previously investigated.

In each of the 28 mixed-mode fracture conditions, three to four test specimens were used, with the exception of one case (notch depth = 3.8 mm) where only one specimen was available for some of the off-axis angles.

The tests were conducted in room temperture on a close-loop Instron tester with a load rate of 1800 Kg/min. The critical load at the onset of the kink crack was recorded on a strip chart. Figs. 4,5 and 6 show the experimental plot of critical laminate stress versus the off-axis angle θ at the onset of the kink crack for specimens of side-notches 2.5 mm, 3.2 mm and 4.5 mm deep, respectively. The case for a = 3.8 mm is not shown because of insufficient numbers of test specimens.

It is seen from the test results that the critical stress, σ_{cr}, at the onset of the kink decreases sharply with the off-axis angle θ when the notch depth is held constant. Similarly, the critical stress also decreases with the increase of the notch depth, a when the angle θ is held constant.

Post-test SEM examination of the fractured surfaces under 500x to 1000x magnifications revealed extensive fiber breaking in the wake of the kink. Fig. 7 presents two such pictures taken near the kink point. Fiber breaks are visible in all cases. It is believed that the observed fiber breakage is due to the good bond between the matrix and the fiber, resulting in fiber nesting and/or fiber bridging accross the kink path.

Finite Element Analysis

The experimental mixed-mode kinking problem is next simulated by the finite element routine. As mentioned earlier, the simulation model must be adjudicated for it's accuracy. In the interest of conciseness, however, details of this development will not be discussed in this paper. Interested readers are referred to Ref. [7].

Return to the off-axis doubly side-notched coupon section shown in Fig. 2. The unidirectional laminate will be assumed an elastic, homogeneous and

orthotropic plate having constants in the principal material coordinates (L,T) determined as follows:

$$E_L = 145 \text{ Gpa} \quad E_T = 10.3 \text{ Gpa} \quad G_{LT} = 6.7 \text{ Gpa} \quad \nu_{LT} = 0.3$$

Now, let the coupon be loaded by the far-field strain, e_x. At some critical value of e_x, the stresses near one of the side-notch tips are assumed to cause a kink emanating from the notch tip and propagate stably in the direction of the fibers. Of interest is when the length of the kink is small compared to the notch depth a. Then, the mixed-mode strain energy release rates G_I and G_{II} at the kink tip are assumed to control the behavior of the initial kink. The values of G_I and G_{II} are calculated by the finite element routine via a crack-closure technique. These can be conveniently expressed in terms of the applied far-field strain in the form:

$$G_I = C_I(e_x)^2 \qquad G_{II} = C_{II}(e_x)^2 \tag{1}$$

where C_I and C_{II} are coefficients from the finite element calculations.

Figs. 8 and 9 show, respectively, the coefficients C_I and C_{II} plotted against the off-axis angle θ, and with the side-notch depth a as an independent parameter. It is seen that the kink is mixed in fracture modes for off-axis angles up to 30^0. Beyond 30^0, the fracture is essentially mode-I. Variation of the mixed-mode ratio, C_{II}/C_I, with the off-axis angle θ is shown in Fig 10. This ratio depends principally on θ, and is almost independent of the notch depth a.

Since for each test coupon the critical stress σ_{cr} at the onset of the kink was measured experimentally. The corresponding critical strain $(e_x)_{cr}$ can be calculated by dividing σ_{cr} by the coupon's axial modulus, E_x. Then, using the values of C_I and C_{II}, the critical strain energy release rates $(G_I)_{cr}$ and $(G_{II})_{cr}$ at

the initial kink for each test case can be calculated via Eq. 1.

For test cases where G_I dominated, the deduced $(G_I)_{cr}$ is clearly G_{IC}. However, for the cases where both mode-I and mode-II were present, a combination of $(G_I)_{cr}$ and $(G_{II})_{cr}$ in some form would control the behavior of the kink. Fig. 11 is a diagram depicting the interactions between $(G_I)_{cr}$ and $(G_{II})_{cr}$ determined from all the test cases.

Though the test data show some degree of scatter, the overall trend indicates that the total strain energy release rate $(G_T)_{cr}$ remain more or less a constant. This strongly suggests that $(G_T)_{cr}$ or G_{TC} essentially controls the behavior of the kink, including the special case of mode-I fracture.

Of course, this suggestion is based only on mixed-mode fracture data with G_{II}/G_I ratios ranging from 0 to about 3. In this range, pure mode-II or predominantly mode-II fracture is not included.

It is also noted that, for graphite-epoxy composites, critical strain energy release rate data for matrix fracture have mostly been limited to G_{IC}. Genrally, the measured values for G_{IC} lie in the range between 120 to 260 J/m^2 depending on the material system used. In this study, G_{IC} has the value in the order of 300 J/m^2. This seems to be on the high side compared to most other accepted values. However, in the present tests, fiber breakage in the wake of matrix cracking was detected in all cases. This could account for the higher measured value for G_{IC}.

CONCLUDING REMARKS

In this paper, mixed-mode matrix fracture in graphite-epoxy composites has been studied using a doubly side-notched, unidirectional off-axis specimen. This specimen has a configuration which is simple to fabricate and versatile in

geometrical variation. As a result, a total of 28 mixed-mode fracture conditions could be produced, which yielded a set of G_{II}/G_{I} ratios covering uniformly from 0 to about 3.

Based on this data, a more definitive conclusion could be reached regarding the criterion for mixed-mode matrix fracture. Specifically, the total strain energy release rate G_{Tc} appears to be a suitable criterion. This criterion, however, may not be applicable to pure mode-II or predominantly mode-II matrix fracture. The latter may involve additional energy dissipating mechanisms such as friction. If so, a separate criterion may be necessary.

REFERENCE

[1] A. S. D. Wang, "Fracture Analysis of Sublaminate Cracks in Composite Materials," Composite Technology Review, Vol. 6, 1984, pp. 45-62.

[2] D. J. Wilkins, J. R. Eisenmann, R. A. Camin, W. S. Margolis and R. A. Benson, "Characterizing Delamination Growth in Graphite Epoxy," in Damage in Composite Materials, ASTM STP-775, 1982. pp. 168-183.

[3] R. L. Ramkumar and J. D. Whitcomb, "Characterization of Mode-I and Mixed-Mode Delamination Growth in T300/5208 Graphite-epoxy," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 315-335.

[4] W. L. Bradley and R. N. Cohen, "Matrix Deformation and Fracture in Graphite Reinforced Epoxies," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 389-410.

[5] A. S. D. Wang, N. N. Kishore and W. W. Feng, "On Mixed-Mode Fracture in Off-Axis Unidirectional Graphite-Epoxy Composites," in Progress in Science and Engineering of Composites, Japan Society for Composite Materials, Vol. 1, 1982. pp. 599-606.

[6] A. J. Russell and K. N. Street, "Moisture and Temperature Effects on the Mixed-Mode Delamination Fracture of Unidirectional Graphite-Epoxy," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 349-370.

[7] W. Binienda, "Mixed-Mode Fracture in Fiber-Reinforced Composite Materials," Ph.D. Thesis, Drexel University, Jan. 1988

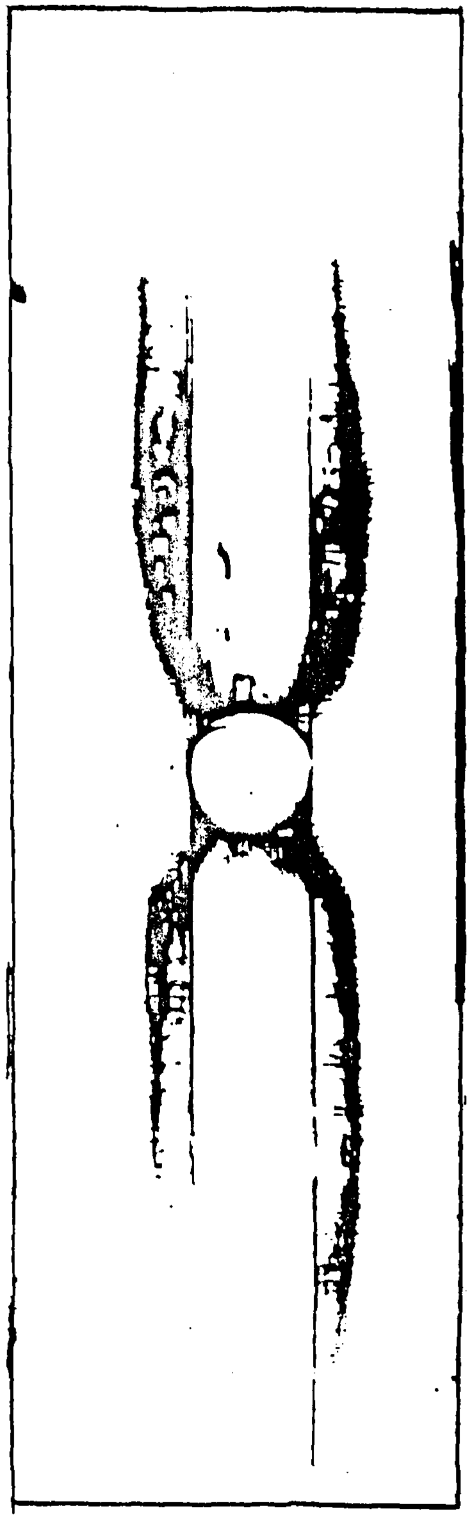


Fig. 1. X-radiograph of matric crack development in a notched $[0_2/90_2]_s$ graphite-epoxy laminate loaded in axial tension

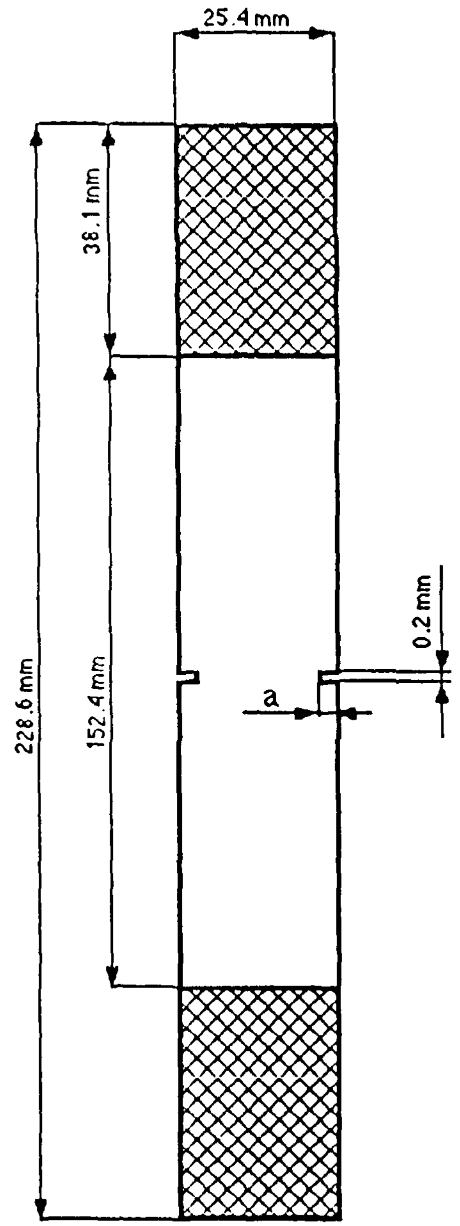


Figure 2 Geometry of the double side-notched specimen.

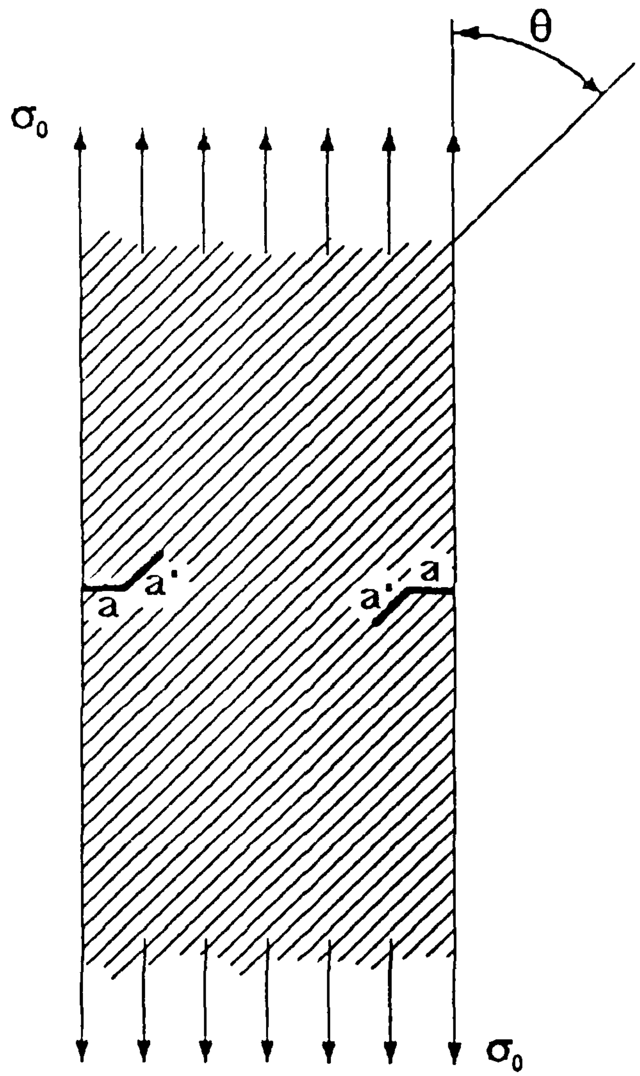


Figure 3 Geometry of kink cracks in the tested specimen.

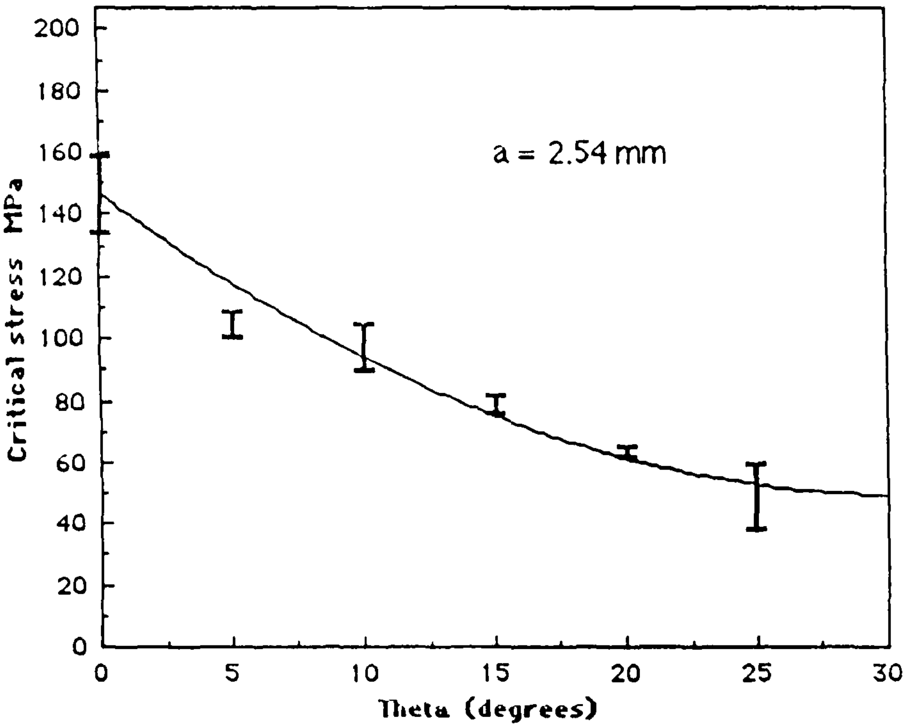


Figure 4 Critical stresses at onset of kink crack. a = 2.54 mm.

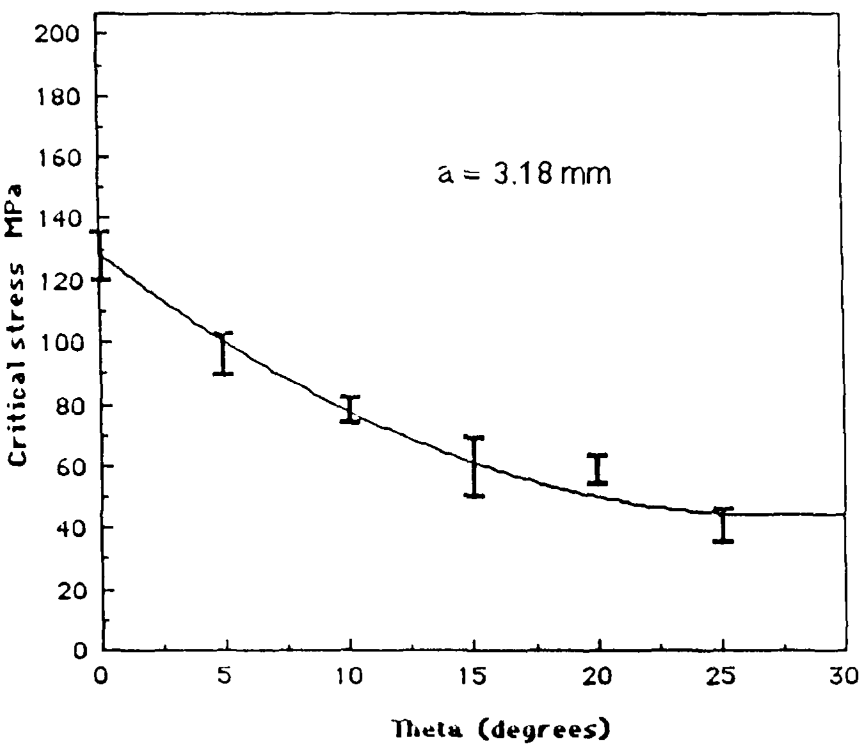


Figure 5 Critical stresses at onset of kink crack. a = 3.18 mm.

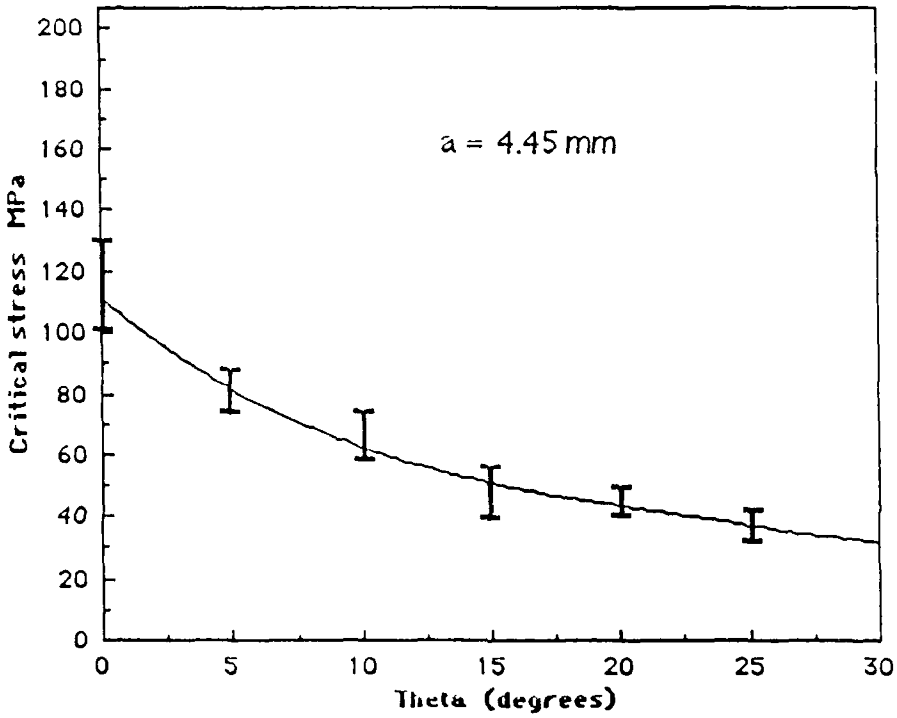


Figure 6 Critical stresses at onset of kink crack. a = 4.45 mm.

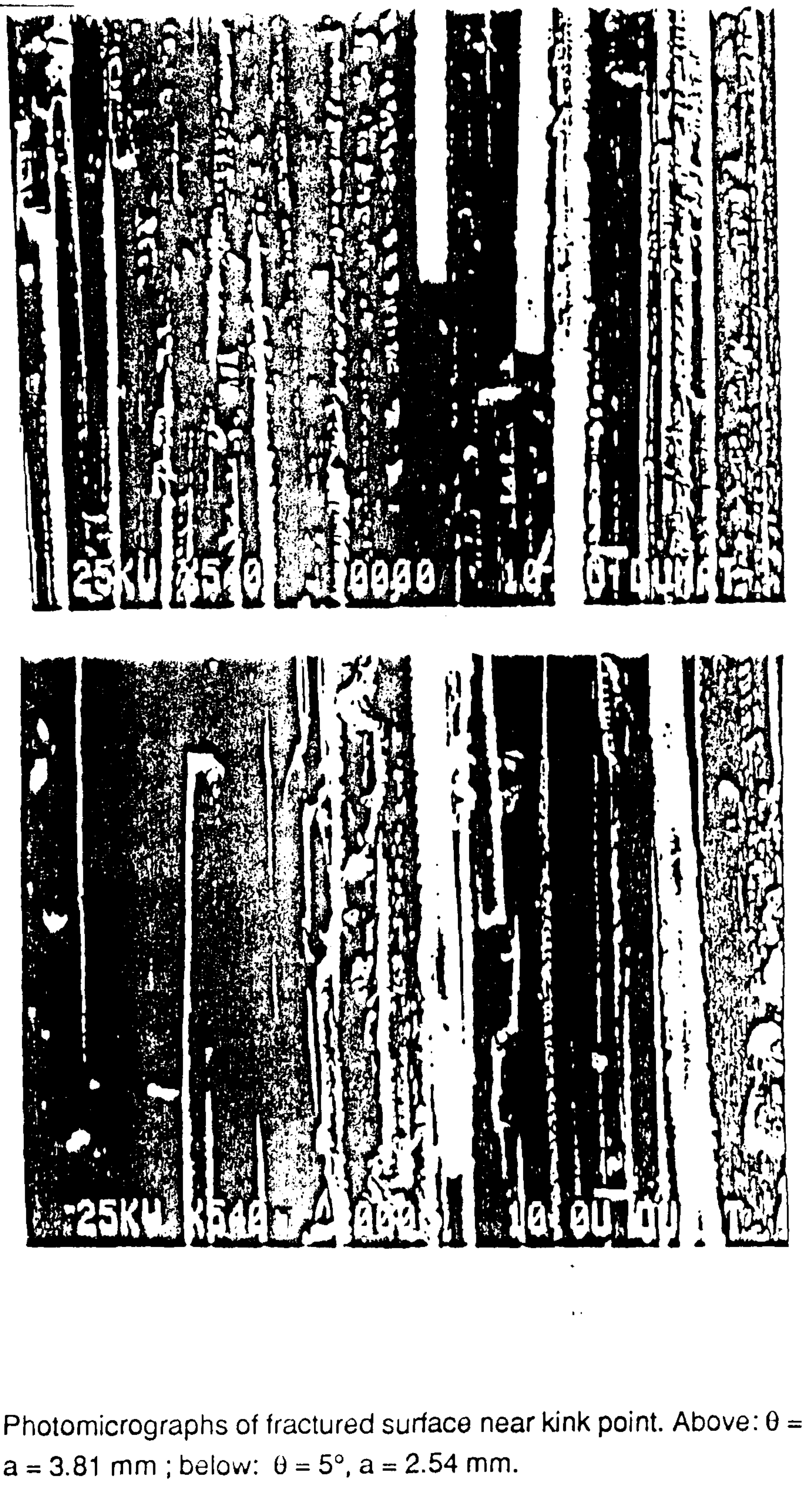


Figure 7 Photomicrographs of fractured surface near kink point. Above: $\theta = 0°$, $a = 3.81$ mm ; below: $\theta = 5°$, $a = 2.54$ mm.

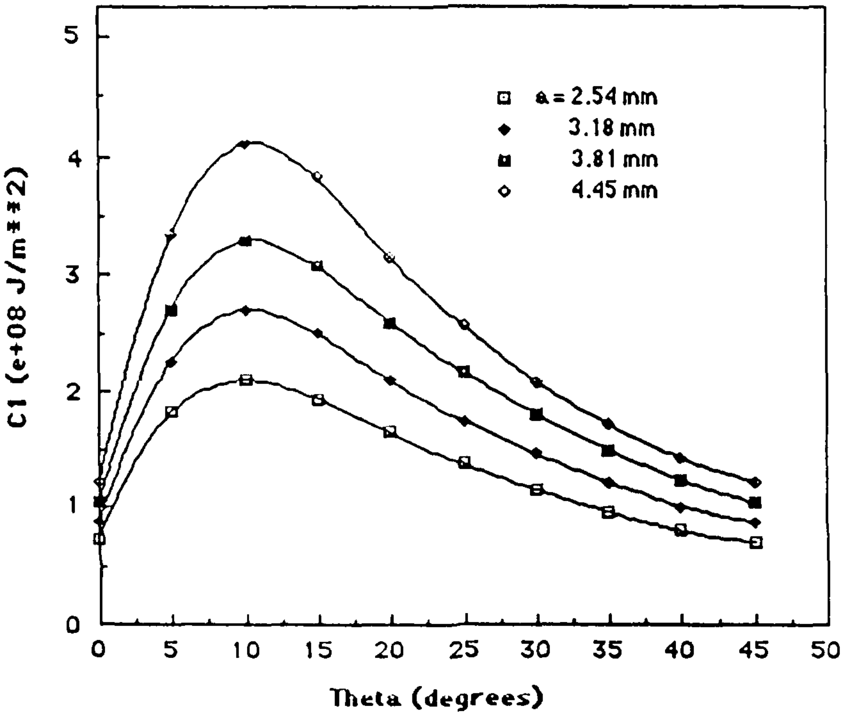


Figure 8 Mode-I strain energy release rate coefficients as function of off-axis angle θ.

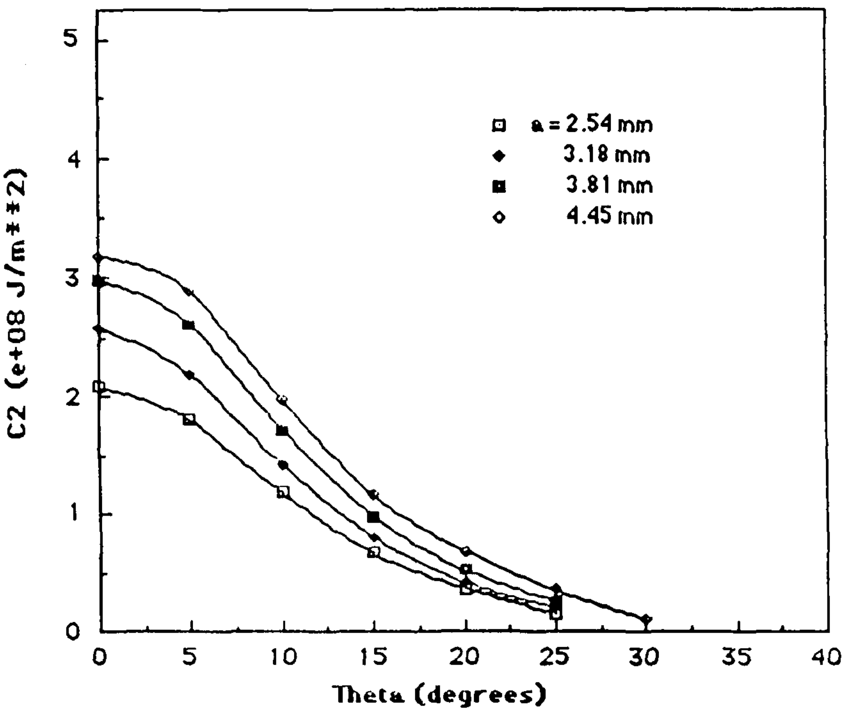


Figure 9 Mode-II strain energy release rate coefficients as function of off-axis angle θ.

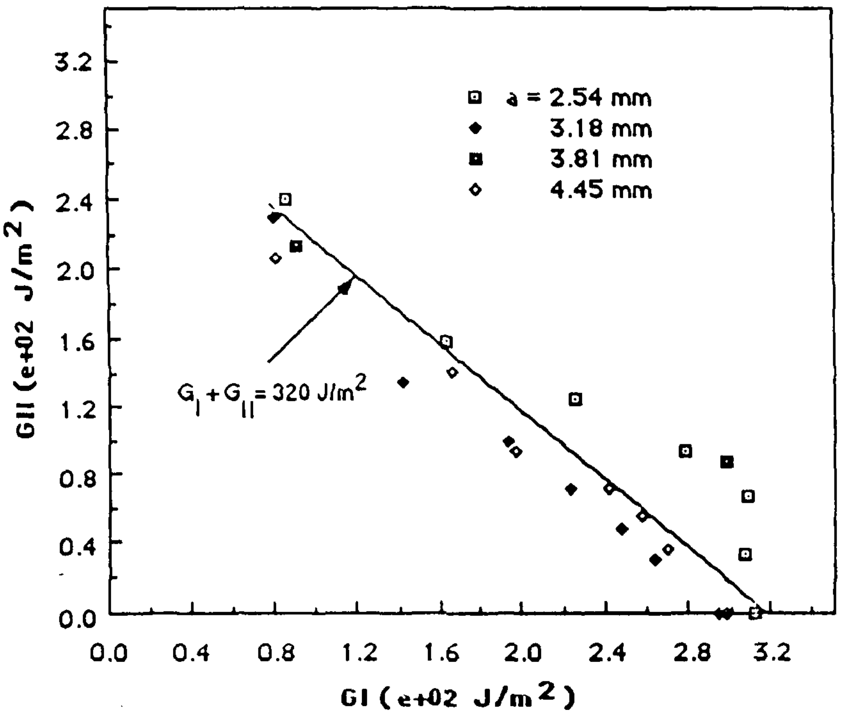


Figure 11 Interaction diagram of mixed-mode strain energy release rate data.

A COMPREHENSIVE STUDY

ON DAMAGE TOLERANCE PROPERTIES OF

NOTCHED COMPOSITE LAMINATES

Appendix III

Three-Dimensional Simulation of Crack Growth in Notched Laminates

Paper presented at the 2nd Annual Meeting, Society for composites, Univ. Delaware; also in Proceedings of the American Society for Composi tes, 1987. pp. 444-457.

Three-Dimensional Simulation of Crack Growth in Notched Laminates

A. S. D. WANG, E. S. REDDY AND YU ZHONG

ABSTRACT

This paper discusses the matrix cracking sequence in a $[0_2/90_2]_s$ graphite-epoxy laminate with double-side notches. Experiments were performed on specimens loaded in uniaxial, quasi-static tension. The specimens were inspected at ascending load increments by x-radiography for patterns of matrix cracks caused by stress concentration near the notched rigion.

A numerical procedure based on a 3-D finite element method was then developed to simulate the observed matrix crack initiation, crack interaction and load-dependent crack growth sequence. The simulation begins with an analysis of the 3-D stress field near the notched region. This is followed by a search of possible modes of matrix cracking and the associated condition for propagation. The concept of brittle fracture is invoked to provide the necessary criterion for identifying the appropriate cracking modes and for determining the associated critical loads for their initiation. A comparison between experiment and prediction is presented.

INTRODUCTION

For a class of structural laminates, initial material damage involves two basic forms of matrix cracking [1]. One form is referred to as intraply cracking where a ply, or a layer of several plies of like fiber orientation, suffers a through-the-thickness crack along the fiber direction. Take the $[0_2/90_2]_s$ laminate coupon under uniaxial tension as an example. A fiber-wise crack in the inner 90^0-layer, known as transverse cracking, is a case of intraply cracking.

A. S. D. Wang, E. S Reddy and Yu Zhong , Professor, Post-doctoral Fellow and graduate student, respectively, Department of Mechanical Engineering and Mechanics, Drexel University, Philadelphia, Pa. 19104

Similarly, a fiber-wise crack in the outer 0^{0}-layer, known as longitudinal splitting, is also a case of intraply cracking. The other basic form is referred to as interply cracking where two adjacent plies suffer a separation in their interface. A delamination in the 0/90 ply interface of the $[0_2/90_2]s$ laminate coupon mentioned above is a case of interply cracking. These two basic forms of matrix cracking may occur independently or interactively, depending on the manner of loading and the lamination structure [2]. Generally, one or both of these cracking modes occur before the load-carrying fibers break.

The initiation and growth mechanisms of intraply and interply matrix cracks, when occuring independently, have successfully been described within the frame work of anisotropic ply elasticity and the fracture theory of brittle cracks [3,4]. A 3-dimensional treatment based on the same analysis concept was recently applied to laminates where the two basic cracking modes occur interactively [5]. In these previous studies, the laminate configuration was that of a straight flat coupon, where free-edge effects dominated the mechanisms.

In this paper, we use laminate coupons with double-side notches to study the formation of interactive matrix cracks that emanate from the notch rather than from the free edge. Since the notch is orientated normal to the applied tension, a very strong stress concentration is induced near the notch-tip. Thus, the intensity of concentration is sentitive to the depth of the notch and alters the matrix cracking characteristics.

Experiments were performed on specimens made of a graphite-epoxy laminate in the form of $[0_2/90_2]s$ tension coupons with side-notches of various depths. For each test specimen, matrix cracking patterns near the notch-tip were inspected by x-radiography at prescribed ascending load increments in order to obtain a load-sequence of the matrix cracking events.

A numerical procedure based on a 3-D finite element method was then developed to simulate the observed load-dependent crack growth. The simulation is based on the strain energy release rate analysis method for non-interactive matrix cracking [3,4] and interactive matrix cracking [5].

A comparisons is made between the predicted load-sequence of events and those recorded experimentally for specimens of different notch depths.

EXPERIMENT

The material used in the experiment was the AS4-3501-06 graphite-epoxy unidirectional system. $[0_2/90_2]s$ laminate panels were made using an autoclave curing procedure. Test coupons were cut from these laminate panels, with dimensions of 25.4 mm wide and 228.6 mm long; the specimen thickness was about 1.016 mm. Double side-notches were introduced at the mid-section

of the coupon by an 8-mil (0.008 in.) diamond saw. Specimens of four notch depths were so prepared (2.54 mm, 3.175 mm, 3.81 mm and 4.445 mm).

Tensile loading was applied to the test specimen through an Instron tester with the cross-head speed set at 0.25 mm per minute. At prescribed ascending load increments, the specimen was x-radiographed at the notched section in order to determine the developing matrix cracking patterns.

For all the specimens tested, the x-radiographs revealed three major forms of matrix cracking during loading. In order of their occurrence, these include longitudinal splitting in the 0^0-layer which emanates from the notch-tip, transverse cracks in the 90^0-layer along with the progression of 0^0-layer splitting and, at some higher load, 0/90 interface delamination growing stably along the length of the 0^0-layer splitting boundary.

Fig. 1 is a sketch of the developing cracking pattern from a specimen with side-notch 3.175 mm deep. It is seen that at the laminate stress of 112 Mpa, a pair of 0^0-layer splits of measurable length emanated from the notch-tip. Initially, the split at one notch-tip grew upward while the split at the other notch-tip grew downward. The growth was extremely stable. At 172 Mpa, splits in four directions emerged from the notch-tips; and a few 90^0-layer transverse cracks appeared between the parallel splits. The 0^0-layer splits grew in length while the 90^0-layer transverse cracks grew in numbers as the laminate stress increased; see sketch corresponding to 259 Mpa. Then, while the splits were still growing, a measurable 0/90 interface delamination initiated along the split boundary near the notch-tip, see sketch corresponding to 319 Mpa. The delamination grew stably as the laminate stress increased; see sketch corresponding to 345 Mpa. The specimen ruptured through the notch section at laminate stress well beyond 600 Mpa.

Fig. 2 is a plot of the measured length of the 0^0-layer split versus the laminate stress, using data from two test specimens having notch depth of 3.175 mm. The scatter in the data is due to variation of the split lengths in four directions. The mean length is taken as the average of the splits in four directions. The laminate stress levels at which 0^0-layer splitting, 90^0-layer transverse cracking and 0/90 interface delamination initiated were all recorded.

Fig. 3 is a plot of the measured 0/90 interface delamination (in area) versus the laminate stress from the same two test specimens. The delamination area at different load increments were measured from prints of x-radiographs using an Lemont Scientific image analyzer. The procedure involves magnification of the delamination area by a high resolution video camera which traverses the contour of the delamination. The scatter of the measured values is due to the variation in areas from the four branches of delamination. From the plot, onset of 0/90 interface delamination may be extrapolated. In this case, delamination onset had occurred at about 260 Mpa.

Table 1 summarizes the onset stresses of the three major forms of matrix cracking from specimens of four notch depths. It is seen that for each form of matrix cracking, onset stress decreases with increase of notch depth. This is expected because the deeper the notch the larger is the stress concentration at the notch-tip.

SIMULATIONS

The Finite Element Model. To simulate the specimen used in the experiment, let us consider the $[0_2/90_2]_s$ laminate having double side-notches at regular interval as shown in Fig. 4a. Assume that these double side-notches are spaced so far apart that they do not interact with one another. Then a periodic element of the laminate which contains only one pair of notches is isolated as shown in Fig. 4b. This element thus represents the test specimen. Note that the laminate is symmetric with respect to the laminate mid-plane (the x-y plane), and the y-axis lies in the plane of the notches. Hence, it is sufficient to model one-eighth of the element shown in Fig. 4b. A schematical finite element mesh is shown in Fig. 4c. Due to expected stress concentration near the notch-tip and ply interfaces, a finer mesh is always deployed in these regions.

The finite element routine was developed based on the assumption that the unidirectional ply is an elastic, homogeneous and orthotropic medium. The elastic and other pertinent material constants for the AS-3501-06 system were characterized by routine tests [6], and their values are listed in Table 2. Solutions for stresses and other quantities, such as strain energy release rates, were obtained by employing a 21-node brick element. The actual computation was carried out on VAX-11/750 and Cray X/MP computers. These and other computational details are found in [7].

Notch-Tip Stress Fields. The laminate stress fields were calculated for two types of loading. The first is by prescribing a far-field laminate strain of $e_x = 10^{-6}$, and the other is by prescribing a uniform temperature change of $\Delta T = -1^0C$. Stresses due to applied laminate tension (by giving a value for e_x) and laminate post-cure cooling (by giving a value for ΔT) can then be obtained by superposition.

Although there are six stress components at each finite element node, it is of interest to examine only those components that are responsible for the observed matrix cracking initiation.

First, let us examine σ_y in the 0^0-layer. This stress is thought to cause 0^0-layer split, which in fact was observed as the first mode of failure under a very low tensile loading. For the case of $e_x = 10^{-6}$, σ_y is tensile throughout the thickness of the 0^0-layer near the notch region. Its value varies from the top to

the bottom of the layer, with the minimum occurring near the 0/90 interface. Fig. 5a shows the σ_y distribution in the 0^0-layer near the 0/90 interface for the specimen having notch depth of 3.175 mm. A sharp rise of σ_y in tension is seen to occur at the notch-tip, displaying a singular behavior. A similarly behaved in-plane shear stress τ_{xy} is also present at the notch-tip; it's planar distribution near the 0/90 interface is shown in Fig. 5b. The concentration intensities of σ_y and τ_{xy} at the notch-tip are about the same.

For the case of $\Delta T = -1^0C$, σ_y is also tensile throughout the 0^0-layer. Fig. 5c shows the σ_y distribution in the 0^0-layer near the 0/90 interface. Here, stress concentration due to the notch is much less. But, by the magnitude of this stress throughout the 0^0-layer is quite large. Thus, the combined tensile and thermal loading will cause the 0^0-layer splitting to be in mixed modes.

Next, let us examine σ_x in the 90^0-layer. This stress causes 90-layer transverse cracking. Again, for the case of $e_x = 10^{-6}$, this stress is tensile and varies throughout the thickness of the 90^0-layer near the notch region, with the minimum occurring near the 0/90 interface and the maximum at the mid-plane. Fig. 6a shows the σ_x distribution in the 90^0-layer near the laminate mid-plane for the specimen having 3.175 mm notch depth. It is seen that a sharp tensile stress is again developed at the notch-tip.

Similarly, for the case of $\Delta T = -1^0C$, σ_x in the 90^0-layer is also tensile with significant magnitude; but stress concentration caused by the notch is minimal, Fig. 6b. Other stress components also exist in the 90^0-layer near the notch-tip; but their magnitudes appear to be negligible.

Finally, the nature of the interlaminar stresses (σ_z, τ_{xz}, τ_{yz}) should be examined because these stresses are responsible for interface delamination. For the same specimen considered under $e_x = 10^{-6}$ loading, its σ_z distribution on the 0/90 interface is shown in Fig. 7a, while distribution on the 90/90 plane is shown in Fig. 7b. It is seen that σ_z can be tensile and of significant magnitude; but it exists only near the notch-tip. As for σ_z caused by thermal cooling, the associated magnitude for σ_z is relatively small. Similarly, the interlaminar shear stresses, τ_{xz} and τ_{yz} also exist with highly localized magnitudes at the notch- tip.

From the above analysis, it appears that 0^0-layer splitting and 90^0-layer transverse cracking are equally likely to occur, while the likelihood for interface delamination is comparatively smaller. However, judgement regarding relative occurrence of these cracking events cannot be made based on the computed stresses, as they all display some degree of stress concentration. In what follows, we attempt to simulate the onset of the observed cracking modes from a fracture point of view.

<u>Simulation of 0^0-Layer Splitting</u>. To simulate the initiation and growth of 0^0-layer splitting, we shall assume that 90^0-layer transverse cracking will

not simultaneously occur. Then, at the notch-tip, we issue a small 0^0-layer split of length s_0 as shown by the insert in Fig. 8. This small split represents an effective flaw which exists at the notch-tip and propagates to become a 0^0-layer split whenever a certain condition is reached. Under a constant far-field strain loading, the split is assumed to propagate stably to reach a length $s > s_0$. Thus, the finite element simulation is to calculate the split-tip stresses and the associated fracture quantity. For the latter, we calculate the split-tip strain energy release rate G as a function of the split length, s.

As was mentioned earlier, the tensile normal stress σ_y and the in-plane shear stress τ_{xy} in the 0^0-layer are the major stress components causing splitting. Fig. 8 is a plot of the split-tip stress σ_y versus the split length, s, for the specimen having 3.175 mm side-notches subjected to $e_x = 10^{-6}$ loading. It is seen that σ_y is larger when s is small, but it deceases sharply with increase of s. On the other hand, the associated shear stress τ_{xy} (not shown) became relatively more dominant with increasing s.

When subject to thermal cooling of $\Delta T = -1^0C$, σ_y in the 0^0-layer is also tensile, see Fig. 5b. But variation of σ_y at split-tip due to growth of split is rather insignificant.

To facilitate a prediction for the load versus split-growth relationship, we then calculate the split-tip strain energy release rate, G(s). This quantity is conveniently expressed in terms of the loads e_x and ΔT [5]:

$$G(s) = [\sqrt{C_e}\, e_x + \sqrt{C_t}\, \Delta T]^2 d \qquad (1)$$

where ΔT represents thermal cooling and d is a length scale which is set at unity in this study. The coefficients C_e and C_t are functions of s and represents the strain energy release rates, corresponding to $e_x = 1$ and $\Delta T = -1^0C$, respectively.

Figs. 9a and 9b show, respectively, the coefficients C_e and C_t versus the split length s for the specimen with 3.175 mm notch depth. It is seen that C_e and C_t both contain mixed modes. However, C_e is predominantly of mode-II, while C_t predominantly of mode-I. When the two load agencies are combined, as in Eq. (1), a mixed-mode cracking of approximately equal ratio results. Note that the overall strain energy release rate is one which decreases with the split length, s; This indicates a stable splitting growth, a behavior consistent with that observed in the experiment.

The load versus split-growth relation is derived from the fracture criterion,

$$G(s) = G_c \qquad (2)$$

where G_c is the critical strain energy release rate for mixed mode cracking. Assume that the value of ΔT is given. Then, by combining Eqs. (1) and (2) we

obtain the critical laminate strain $(e_x)_{cr}$ as a function of split length, s.

For the material system used in this study, ΔT and the mixed mode G_c have been determined elsewhere [6]; and their values are listed in Table 2. Thus, the predicted $(e_x)_{cr}$ can be converted to $(\sigma_x)_{cr}$. For the specimen just considered, the computed $(\sigma_x)_{cr}$ versus s relations is shown by the solid line in Fig. 2. It is seen that the predicted result agrees well with the initial portion of the experimental split-growth data, where the splitting was not yet significantly complicated by the development of 90^0-layer transverse cracks. The predicted curve, however, departs away from the observed results as 90^0-layer transverse cracks developed in higher density. To include the effects of these transverse cracks on split growth will require a major modification of the simulation model.

The predicted onset stresses for 0^0-layer splitting for specimens of four notch depths are listed in Table 1 along with their experimental counterparts. In all cases, the model seems to predict well the initiation of the splitting.

<u>Simulation of 0/90 Interface Delamination.</u> Delamination of the 0/90 interface takes place at much higher load. Once initiated, it grows stably along the boundary of the 0^0-layer splits. The delamination pattern is shown schematically in Fig. 10. As we have observed in the experiment, 90^0-layer transverse cracks actually formed continuously as the 0/90 interface delamination grew, see Fig. 1. An analytical/computional simulation of this complex interactive cracking phenomenon, though not impossible, is quite tedious and probably not fruitful. Thus, a simplified version is attempted instead. Namely, we shall assume that only 0^0-layer splitting procedes the initiation of the 0/90 interface delamination and the effect of 90^0-layer transverse cracking is negligible.

The simulation follows a similar procedure as used for 0^0-layer splitting. A set of densely meshed finite elements is deployed near the intended delamination region, double nodes are assigned on the plane of delamination and these are then released in sequence so as to mimick the actual growth pattern observed in the experiement. In the node-releasing process, the fracture energy release rate coefficents, C_e and C_t are computed as functions of the delaminated area [7]. Figs. 11a and 11b show, respectively, the computed C_e and C_t coefficients versus delamination area for the specimen having the notch depth of 3.175 mm.

From the energy release rate curves, it is seen that the delamiantion is primarily of mode-III under the applied tenile loading (e_x=1), while primarily in mode I and II under thermal cooling (ΔT=-1^{0}C). Thus, the combined effect is again one of mixed modes. The overall energy release rate, however, decreases sharply with increasing delamination area, indicating a stable growth. This is

also consistent with the behavior observed in the experiment.

Using the computed energy release rate coefficients, stress levels corresponding to the prescribed delamination node-releasing sequence can be predicted by means of Eqs. (1) and (2). For example, in the case of the specimen having notch depth of 3.175 mm, the predicted delamination growth curve is shown by the solid line in Fig. 3. Here, again, the agreement between prediction and experiment is quite close for the initial portion of the delamination growth. Apparently, as the delamination grows larger, many transverse cracks are formed in the 90^0-layer; the associated cracking mechanisms then becomes more complicated than the model has portrayed.

The critical stresses for 0/90 delamination in specimens having other notch depths were also computed. These are listed in Table 1 for comparison with their experimental counterparts.

CONCLUSIONS

In this paper, we have presented a method of simulation for matrix cracks that develop in laminate specimens having double side-notches. The analysis entails a 3-D stress analysis and computer simulation of fracture growth near the notched region. The purpose of the study is to understand the damage mechanisms at a level below the lamination structure. The actual specimen chosen for analysis could represent a critical element [8] in a large laminated structure whose global strength and/or fatigue properties are to be evaluated.

Acknowledgments: Results reported in this paper were obtained during the course of research supported by a grant from the Air Force Office of Scientific Research.

REFERENCES

1. A. S. D. Wang, "Fracture Mechanics of Sublaminate Cracks in Composite Laminates," Composite Technology Review, Vol. 6, 1984. pp. 45-62.

2. F. W. Crossman and A. S. D. Wang, "The Dependence of Transverse Cracking and Delamination on Ply Thickness in Graohite-Epoxy Laminates," in Fatigue of Fibrous Composite Materials, ASTM STP-775, 1982. pp. 118-139.

3. A. S. D. Wang and F. W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1: An Energy Method," Jour. Composite Materials, Vol. 14, 1980. pp. 71-87

4. F. W. Crossman, W. J. Warren, A. S. D. Wang and G. E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2: Experimental Correlation," Jour. Composite Materials, Vol. 14, 1980. pp. 88-108.

5. A. S. D. Wang, N. N. Kishore and C. A. Li, "On Crack Development in Graphite-Epoxy $[0_2/90_{2n}]_s$ Laminates Under Uniaxial Tension," Composite Science and Technology, Vol. 24, 1985. pp. 1-31.

6. A. S. D. Wang, W. Binienda and Y. Zhong, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites," NADC Technical Report, 1987.

7. A. S. D. Wang, E. S. Reddy and N. N. Kishore, "A three Dimensional Finite Element Code and Users Manual for Analysis of Matrix Cracking in Laminates," Drexel University Technical Report, 1987.

8. K. L. Reifsnider, "Critical Element Concepts for Residual Strength and Life Prediction," Proc. American Society for Composites, Technomic Publishing, 1986. pp. 387-403.

Table 1. Experimental and Predicted (in parenthesis) Onset Stresses

Notch Depth	2.54 mm	3.175 mm	3.81 mm	4.445 mm
0^0-layer Split	100 Mpa (80 Mpa)	75 Mpa (70 Mpa)	60 Mpa (60 Mpa)	60 Mpa (60 Mpa)
90^0-layer Transverse Crack	170 Mpa	160 Mpa	150 Mpa	150 Mpa
0/90 interface Delamination	350 Mpa (300 Mpa)	260 Mpa (230 Mpa)	225 Mpa (200 Mpa)	220 Mpa (180 Mpa)

Table 2. Pertinent Material Constants for AS4-3501-06 UD Ply

$E_{LL} = 145$ Gpa $E_{TT} = E_{ZZ} = 10.3$ Gpa $G_{LT} = G_{LZ} = 6.8$ Gpa $G_{TZ} = 3.5$ Gpa

$\nu_L = \nu_{LZ} = 0.3$ $\nu_{TZ} = 0.54$ $\alpha_L = 0.4\times10^{-6}/^0C$ $\alpha_T = \alpha_Z = 28.8\times10^{-6}/^0C$

$\Delta T = -140\ ^0C$ $(G_c)_{total} = 289\ j/m^2$ Ply Thickness = 0.127 mm

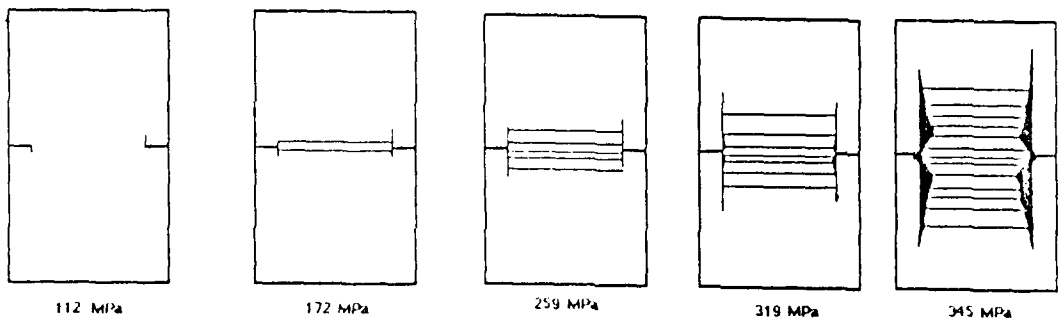


Fig. 1. Development of Matrix Cracks in Specimen of Notch Depth 3.175 mm

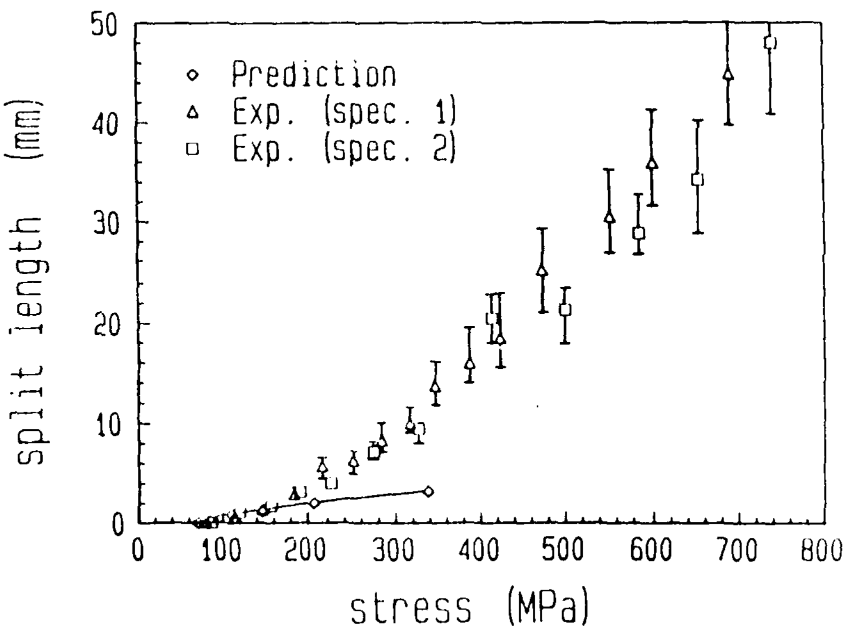


Fig. 2. Split Length Growth versus Applied Tension (Notch Depth, 3.175 mm)

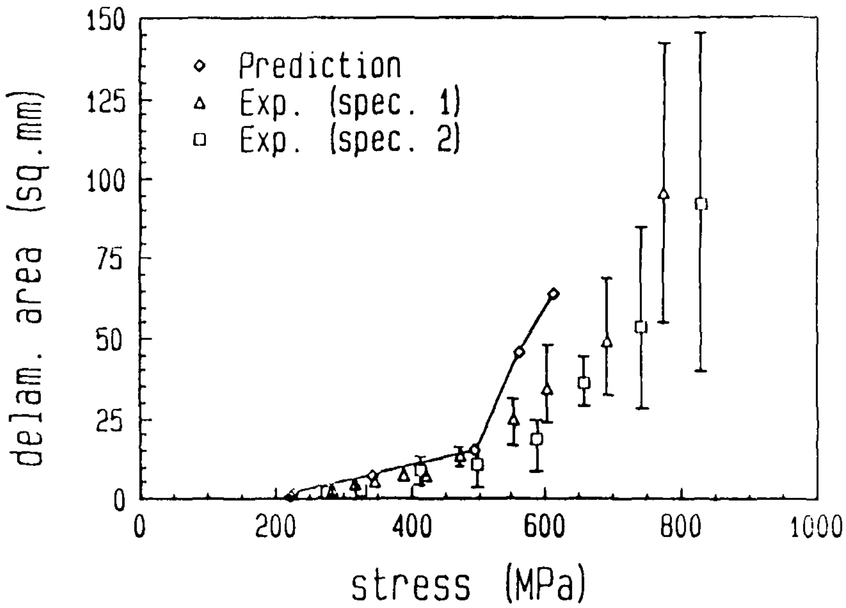


Fig. 3. Delamination Area Versus Applied Tension (Notch Depth 3.175 mm)

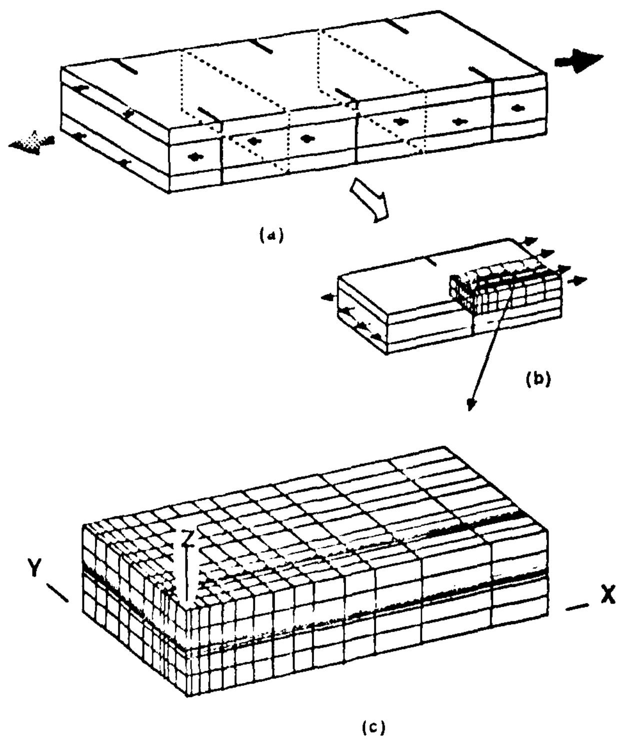
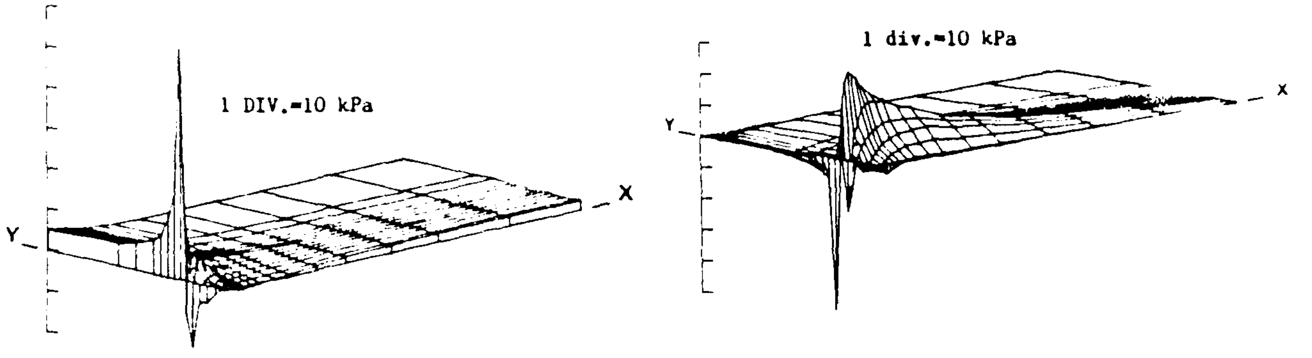
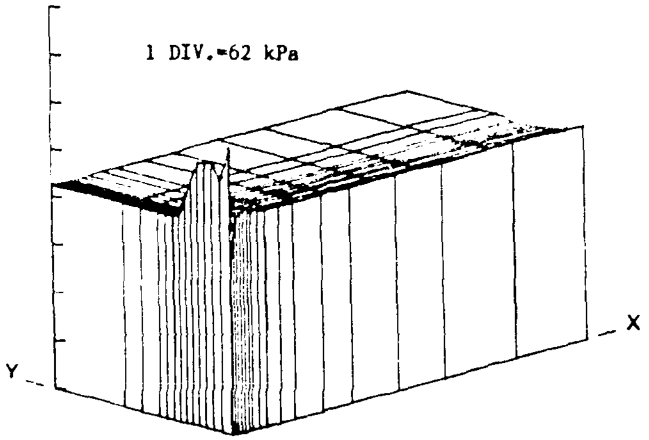


Fig. 4. Finite Element Model for the Double Side-Notched Specimen.

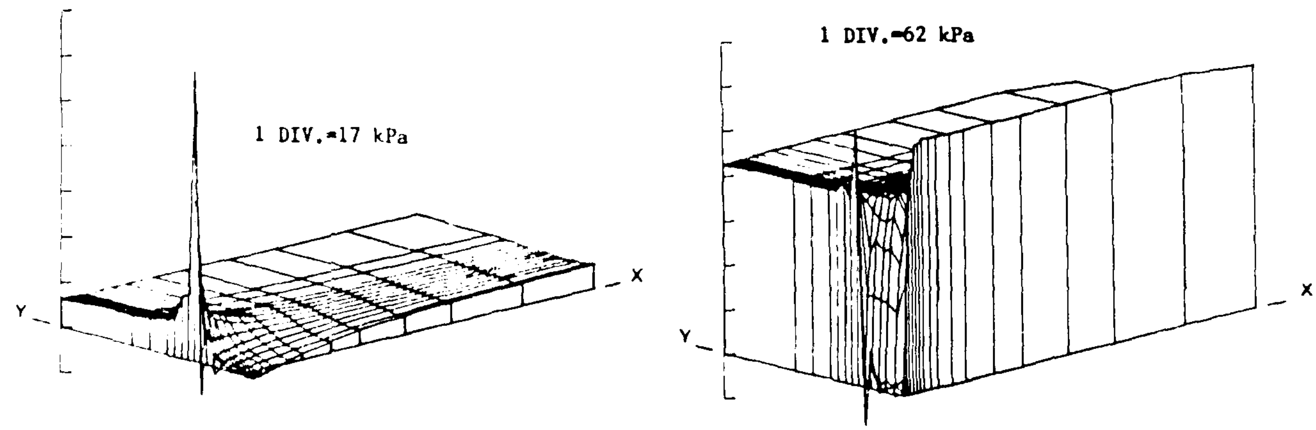


(a) (b)



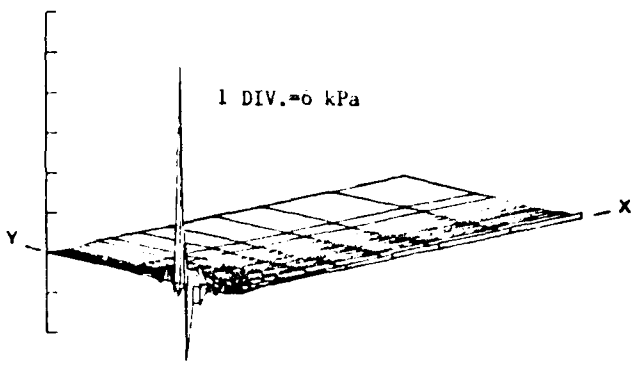
(c)

Fig. 5. (a) σ_y Distribution in the 0^0-Layer ($e_x=10^{-6}$)
(b) τ_{xy} Distribution in the 0^0-Layer ($e_x=10^{-6}$)
(c) σ_y Distribution in the 0^0-Layer ($\Delta T=-1^0C$)

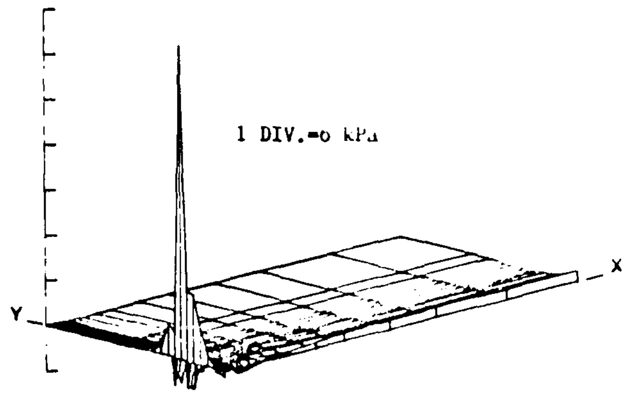


(a) (b)

Fig 6 σ_x Distribution in 90^0-Layer due to (a) $e_x=10^{-6}$ and (b) $\Delta T=-1^0C$



7(a)



7(b)

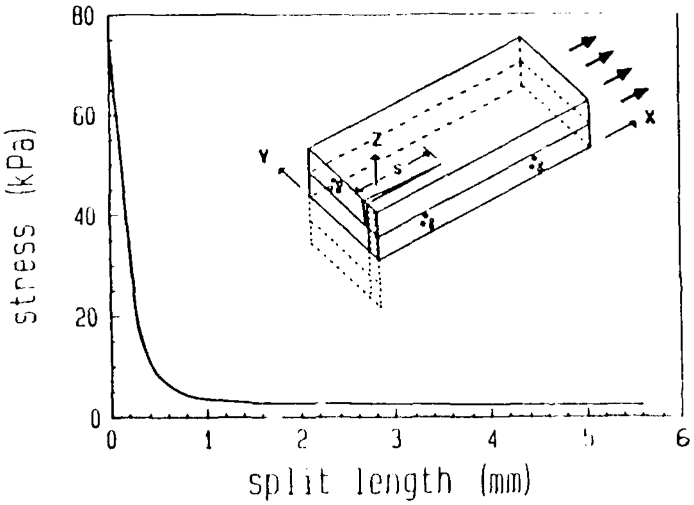
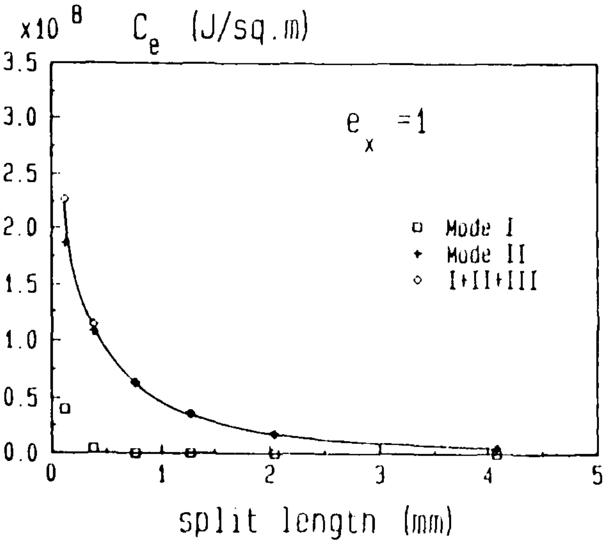


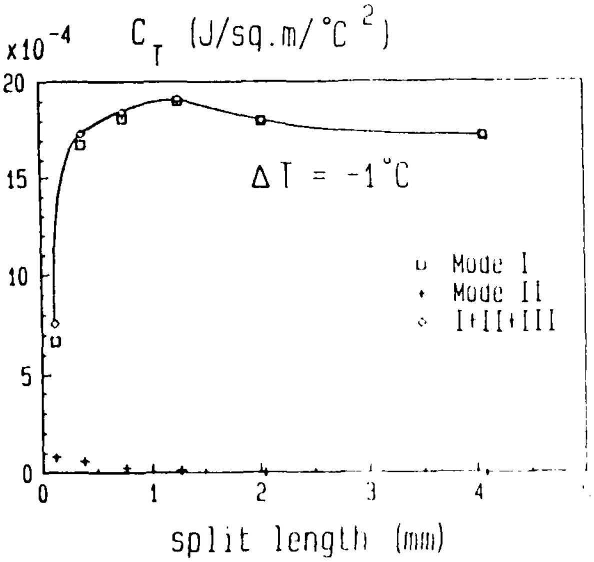
Fig. 7. σ_z Distribution on (a) 0/90 and (b) 90/90 Interface

Fig. 8. σ_y at Split-tip versus Split Length, s

Fig. 9. Energy Release Rates at Split-tip. (a) C_e and (b) C_t



9(a)



9(b)

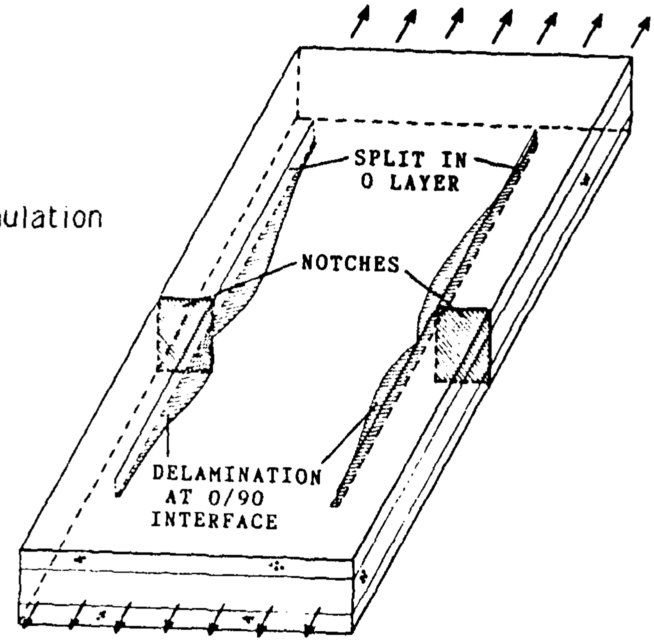


Fig. 10. Finite Element Simulation of 0/90 interface Delamination.

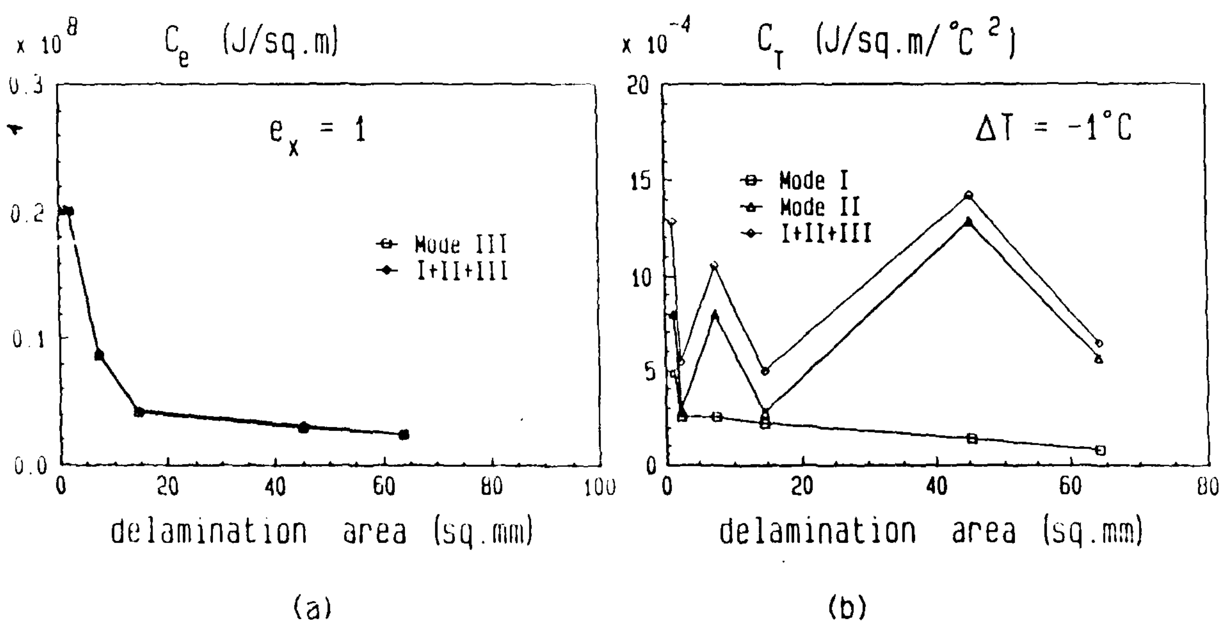


Fig. 11. Energy Release Rate at Delamination Front: (a) for $e_x = 1$, (b) $\Delta T = -1^{0}C$

A COMPREHENSIVE STUDY

ON DAMAGE TOLERANCE PROPERTIES OF

NOTCHED COMPOSITE LAMINATES

Appendix IV

Simulation of Matrix Cracks in Composite Laminates Containing a Small Hole

Paper presented at the ASME Winter Annual Meeting, Boston, 1987;
Also in Damage Mechanics in Composites, AD-12, ASME, 1987. pp. 83-91.

SIMULATION OF MATRIX CRACKS IN COMPOSITE LAMINATES CONTAINING A SMALL HOLE

E. S. Reddy, A. S. D. Wang, and Y. Zhong
Drexel University
Philadelphia, Pennsylvania

ABSTRACT

This paper studies the matrix cracking sequence in $[0_2/90_2]_s$ graphite-epoxy laminates that contain a small central hole. Experiment was performed first using specimens loaded in uniaxial, quasi-static tension, followed by inspections of the specimen at several prescribed loading increments by means of x-radiography. The inspection provides a quantitative measurement and a physical analysis of matrix cracking patterns near the hole.

A numerical procedure based on a three dimensional finite element method was then employed to simulate the observed matrix cracking patterns, including their initiation and growth behaviors. Here, the theory of ply elasticity and the concept of brittle fracture are used as basis for the finite element simulation.

A comparison between the simulated and the experimental results is presented.

1. INTRODUCTION

Failure analysis of fiber-reinforced composites has attracted increased interest in recent years. Application of composites in high-performance aircraft and spacecraft structures has led the researchers to carry out intensive experimental and theoretical studies on failure mechanisms in a variety of composite materials. For a special class of composites, namely, polymeric laminates made by laminating unidirectional continuous fiber systems, failure initiation usually involves some forms of matrix cracking. When viewed at the laminating ply level, these can be classified into two basic forms. One basic form is known as intraply cracking, where a ply or several plies of the same fiber orientation that formed a layer, suffers a through-the-thickness crack along the fiber direction. A simple example of intraply cracking is found in a $[0/90]_s$ type laminate under axial tension, in which the inner 90^o-layer suffers multiple transverse cracks. The other basic form is interply cracking, where two adjacent plies in the laminate suffer a separation in their interface. Free edge delamination in a $[\pm45/0/90]_s$ laminate loaded in axial tension, for *instance, provides* a case of interply cracking.

Studies of the individual growth mechanisms of the two basic forms of matrix cracking have been extensively reported in the literature (see, e. g. [1,2]). Interactions of the two basic forms of cracking were examined partially in [3,4]. The problems studied in [1-3] concerned cracking development in plain laminates, while the problem studied in [4] involved laminates that contain sharp through-the-thickness notches.

The problem of laminates with a through-hole has also attracted considerable attention. Effects of ply stacking sequence [5] and different material combinations [6] on global laminate strength reduction due to presence of a small hole were among the early interests. Subsequent analyses have focussed on the detailed stress distribution around the hole, especially the interlaminar stresses that cause local delamination [7-9]. In these works, delamination (interply cracking) is assumed to take place as the only matrix cracking mode. Experiments using graphite-epoxy laminates have shown, however, that the first matrix cracking form near the hole is usually not delamination.

In the present study, we use a $[0_2/90_2]_s$ graphite-epoxy laminate with a small central hole to examine the initiation and growth patterns of matrix cracks near the hole. In this case, a three dimensional stress analysis based on ply elasticity *performed*, which shows that severe stress concentrations along the hole boundary are present, and matrix cracks of different forms may initiate and propagate at these locations. At the same time, experiments performed on test specimens and inspected by x-radiography at different loading levels reveal the exact sequence of the various cracking events. Thus, the purpose of this study is to relate the experimental events with the analysis by means of a finite element simulation. A comparison is then made between the simulated and the experimental results.

2. EXPERIMENT AND RESULTS

In the experiment, test coupons were made from AS4-3501-06 graphite-epoxy prepreg tapes. The lamination stacking sequence was limited to $[0_2/90_2]_S$.The dimensions of the test coupons were 25.4 mm wide, 228.6 mm long and 1.016 mm thick. The radius of the central hole was 3.175 mm. Loading was applied axially on an Instron tester, with the cross-head speed set at 0.25 mm per minute. The loaded specimens were periodically inspected by DIB enhanced x-radiography.

Experimental results show three major forms of matrix cracking that emanate from the hole during loading. In their order of occurrence, these are (1) horizontal transverse cracks (intraply cracking) in the inner 90^0-layer in the immediate region of the hole initially, and away from the hole region subsequently; (2) vertical splitting cracks (also a form of intraply cracking) in the outer 0^0-layers emanating from the hole and propagating stably away from the hole; and (3) delamination (interply cracking) in the 0/90 interface along the length of the 0^0-layer splits, which displays a very stably growth behavior. Figure 1 is a schematic illustration of the cracking development patterns at five typical laminate stress levels. It can be seen that a few (two or three) 90^0-layer transverse cracks and an initial sign of 0^0-layer splits are present at about 276 MPa. At 379 MPa, four branches of 0^0-splitting have already formed and propagated stably towards the top and bottom of the specimen. Note that propagation of the splits are accompanied by more 90^0-layer transverse cracks, see illustration at 465 MPa. At 552 MPa, delamination in the 0/90 interface has already occurred along each of the four 0^0-layer splits. While the delamination grows stably with load, more transverse cracks have formed and the longer the 0^0-splits have grown, see illustration at 724 MPa. At this load level, matrix-related damage around the hole is substantial; but no significant fiber breakage has yet occurred. In fact, the specimen can sustain a laminate stress of more than 1000 MPa before it breaks completely through the hole section.

To express quantitatively the observed matrix cracking, we choose to display two separate cracking quantities in terms of the applied laminate stress. The first is the linear length of the 0^0-layer split and the second is the area of the 0/90 interface delamination. Since for each specimen there are four branches of splits, which grow stably with load, a mean length is obtained from measuring all four splits at each load interval. Figure 2 shows the mean split length plotted against the laminate stress (in MPa), where the data are from a sample of six specimens. It is seen that the mean split grows almost linearly with the applied laminate stress; and by extrapolation of the data, we can deduce that the onset stress for 0^0-layer split is at about 120 MPa. The solid line in the figure represents the simulated split growth. The adequacy of the simulated result will be discussed in the next section.

Similarly, Figure 3 shows the mean delamination area measured from the same six specimens (the areas were measured from the x-radiographs using an image analyzer). Here again, we see that the delamination growth rate is quite slow initially but becomes rapid as the laminate stress is increased. Data extrapolation yields the onset stress at about 220 MPa. The solid line in this figure is the simulated results. However, we shall defer the discussion on simulation later in the next section.

3. SIMULATIONS AND RESULTS

The Finite Element Model

The problem of a laminate with a single central hole stems from the large composite structural panel with bolt or rivet holes. In these large structural laminates, the holes are often periodically placed. Assuming that the holes are located so far apart that they do not interact with each other, then a periodic element of the panel containing only one hole can be considered for analysis, see Figure 4a. For the problem considered here, the lay-up of the specimen is $[0_2/90_2]_S$ and the hole is placed at the laminate center. Thus, it is sufficient to model one-eighth the specimen due to symmetry as shown in Figure 4b. Since stress concentration is expected around the hole boundary, a finer finite element mesh is deployed in this region in order to capture the true nature of stress concentration.

It is noted that the basis of the finite element analysis is the assumption of ply elasticity, that is that the graphite-epoxy unidirectional ply is assumed as an elastic, homogeneous and orthotropic medium. The elastic and the thermal expansion constants of the AS-3501-06 ply system were characterised and given in [11]. The basic finite element is a 21-node solid brick and the computation is performed on a CRAY X/MP computer. Details of the computational procedures are contained in a separate user's manual [12].

Stresses Near the Hole Boundary

The laminate stress fields are calculated for two types of loading. the first is by prescribing a far-field laminate strain of $e_x = 10^{-6}$, and the other is by prescribing a uniform temperature change of $\Delta T = -1^0C$. Stress due to combined tension and temperature change can be obtained by superposition. The stress field near the hole boundary is in a complicated three dimensional state. It is not of interest here to examine all of the stress distributions in detail. Rather, we shall display only some typical ones that are thought to cause matrix cracking.

Figure 5 is a display of the xy-plane distribution for the six stress components which exist in the 0^0-layer near the 0/90 interface. Here, the largest stress is σ_x which is in the fiber direction. But when compared to the ply strength in the fiber direction this stress is rather insignificant in causing failure. Other stresses that may cause matrix failure are σ_y and τ_{xy}. These two combined can cause longitudinal splitting in this layer. The interlaminar shear stresses τ_{zx} and τ_{zy} are relatively small but could precipitate 0/90 interface delamination.

Figure 6 is a similar display for the stresses in the 90^0-layer near the mid-plane of the laminate.Here, we see the dominance of σ_x which is normal to the fibers in this layer. Concentration near the hole region will be certain to cause transverse cracks. All other stresses however have secondary influence.

The above stress distributions are computed for tension

loading only. We also need the stress distribution due to thermal cooling of the laminate in order to evaluate the combined stress state. For the laminate used, ΔT is set at -140 ^oC. For simplicity however, we shall omit the display of the thermal stresses here.

From the stress analysis, it appears that 0^o-layer splitting and 90^o-layer transverse cracking are equally likely to occur first. However, because of the high stress gradient near the hole boundary it is not possible to make a prediction as to which of these two forms of matrix cracking will first occur. In what follows, we attempt to simulate the onset and propagation of some of these cracking forms from a fracture point of view.

Simulation of 0^o-Layer Splitting

In the simulation of 0^o-layer splitting, it is assumed that 90^o-layer transverse cracks are absent while the split grows with loading. This assumption is necessary to reduce the geometric complexity of the cracked laminate. It is felt that omission of the transverse cracks will not adversely affect the accuracy of the simulation, at least not the onset of splitting. To simulate, a small split length s_o is introduced in the 0^o-layer as shown by the inserted sketch in Figure 7. This small split represents an effective material flaw which exists at the hole boundary and propagates to become a split whenever the critical condition is reached. Under the far-field constant strain loading, the split is assumed to propagate stably in the fiber direction. Thus, the finite element routine is to calculate the stresses and the strain energy release rate G at the split-tip as a function of split length s.

Figure 7 shows the variation of σ_y at the split-tip with the split length s. It is believed that this stress is responsible for the initiation and continuation of the split. From the figure, we see that σ_y is large when s is small; but it decreases sharply with increase of s. On the other hand, the associated split-tip shear stress τ_{xy}, which is not shown here, becomes relatively more dominant with increase in s. This indicates that once the split starts, it will propagate stably and in shearing mode.

The corresponding stresses due to thermal cooling are also calculated; their effect on splitting is included in the prediction, which is to be discussed below.

As mentioned, we first introduce a small split length s_o and then calculate the split-tip strain energy release rate G(s) as a function of $s \geq s_o$. G(s) can be expressed in terms of the applied tension e_x and thermal loading ΔT as [5]:

$$G(s)_i = \left\{ \left[\sqrt{C_e} e_x + \sqrt{C_T} \Delta T \right]^2 d \right\}_i \qquad i = I, II, III \qquad (1)$$

where the coefficients C_e and C_T are functions of s and represent the strain energy release rates corresponding to $e_x = 1$ and $\Delta T = -1$ ^oC, respectively. The parameter d is a length scale factor which is set at unity in this study. Finally, the subscript i refers to cracking modes of I, II and III (open, sliding and antiplane shear).

Now, Figures 8 and 9 show, respectively, the coefficients C_e and C_T versus the split length s. Note that C_e is predominantly of mode II, while C_T is predominantly of mode I. Thus, the combined crack growth is in mixed mode.

The growth behavior of mixed mode matrix cracking is discussed in [11] and a criterion based on the total energy release rate is suggested:

$$\sum G(s)_i = G_c \qquad (2)$$

where G_c is the total critical strain energy release rate for mixed mode cracking. For the material used here, G_c has a value estimated at 289 J/m^2.

By using the coefficient curves in Figures 8 and 9, we can obtain from (1) and (2) the critical laminate strain e_x as a function of split length s. The computed critical laminate strain can be readily converted in laminate stress; the stress versus s relation is shown in Figure 2 by the solid line. It is seen that the calculated result agrees initially with the experiment, predicting the onset of splitting. As the split grows longer, the splitting mechanisms are complicated by transverse cracking and also by delamination. Since these complicating mechanisms are not included in the splitting simulation model, a discrepancy between the experiment and the prediction results.

Simulation of 0/90 Interface Delamination.

In the simulation of the 0/90 interface delamination, an idealization is also made. Namely, we assume that delamination occurs after the 0^o-layer splitting has grown a sufficient length so that delamination is proceeding as an independent event. The simulation is carried out to mimic the delamination shape as observed in the experiment. The simulated shape is schematically shown in Figure 10. Here, we calculate the mean strain energy release rate coefficients at the delamination front as a function of the total delaminated area, see Figures 11 and 12. Then, by means of the criterion in (2) we obtain the delamination area versus laminate stress relation as shown in Figure 3 by the solid line. Again, the prediction for the onset of delamination is close, but discrepancy results once the delamination has grown larger. This is expected because the actual growth of delamination is concurrent with other forms of matrix cracking, as discussed earlier in the experimental study. This complex mechanics mechanisms was not included in the simulation model.

4. CONCLUSIONS

In this paper, we have shown that growth of matrix cracks in the vicinity of a small hole in a laminate can be reasonably simulated by a finite element routine based on a careful fracture mechanisms analysis. Still, the actual mechanisms are complicated and the simulation has to resort to some degree of idealization. This causes discrepancies between the simulated results and the experiment. It is conceivable that these discrepancies can be considerably removed if more is known about the physics of the phenomenon at the microscopic level and if a more powerful simulation technique is available.

Acknowledgments: Results reported in this paper were obtained during the course of research supported by a grant from the Air Force Office of Scientific Research.

REFERENCES

[1] A.S.D. Wang and F.W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1, An Energy Method", J.Composite Materials, Vol. 4, pp. 71-87, 1980.

[2] F.W. Crossman, W.J. Warren, A.S.D. Wang and G.E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2, Experimental Correlation", J. Composite Materials, Vol. 14, pp. 88-106, 1980.

[3] A.S.D. Wang, N.N. Kishore and C.A. Li, "Crack Development in Graphite-Epoxy Cross-ply Laminates Under Uniaxial Tension", Composites Science and Technology, Vol. 24, pp. 1-31, 1985.

[4] A.S.D. Wang, E.S. Reddy and Yu Zhong, "Three Dimensional Simulation of Crack Growth in Notched Laminates", to be presented at Joint Symposium on Composite Materials Science and Engineering, American Society for Composites - University of Delaware Center for Composite Materials, September 23-25, University of Delaware, Newark DE, 1987.

[5] E.F. Rybicki and D.W. Schmueser, "Effect of Stacking Sequence and Lay-up Angle on Free Edge Stresses Around a Hole in a Laminated Plate Under Tension", J. Composite Materials, Vol. 12, pp. 300-313, 1978.

[6] I.M. Daniel, R.E. Rowlands and J.B. Whiteside, "Effects of Materials and Stacking Sequence on the Behavior of Composite Plates with Holes", Experimental Mechanics, Vol. 14, pp. 1-9, 1974.

[7] T.K. O'Brien and I.S. Raju, "Strain Energy Release Rate Analysis of Delamination Around an Open Hole in Composite Laminates", Paper No. AIAA-84-0961, AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics & Materials Conference, May 14-16, Palm Springs Ca., 1984.

[8] I.S. Raju and J.H. Crews, "Three-Dimensional Analysis of $[0/90]_S$ and $[90/0]_S$ Laminates with a Central Circular Hole", Composite Technology Review, Vol. 4, No. 4, pp. 116-124, 1982.

[9] L. Carlsson, "Interlaminar Stresses at a Hole in a Composite Member Subjected to a In-plane Loading", J. Composite Materials, Vol. 17, pp.241, 1983.

[10] K. Ericson, M. Persson, L. Carlsson and A. Gustavsson, "On the Prediction of the Initiation of Delamination in a $[0/90]_S$ Laminate with a Circular Hole", Journal of Composite Materials, Vol. 18, pp. 495-506, 1984.

[11] A.S.D. Wang, W. Binienda, Yu Zhong and E.S. Reddy, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites", NADC-TR-87133-60, Naval Air Development Center, Warminster, Pa. 18974, April 1987.

[12] E.S. Reddy, N.N. Kishore and A.S.D. Wang, "A Three Dimensional Finite Element Code and Users' Manual for Matrix Cracking in Laminates", Drexel University Technical Report, September 1987.

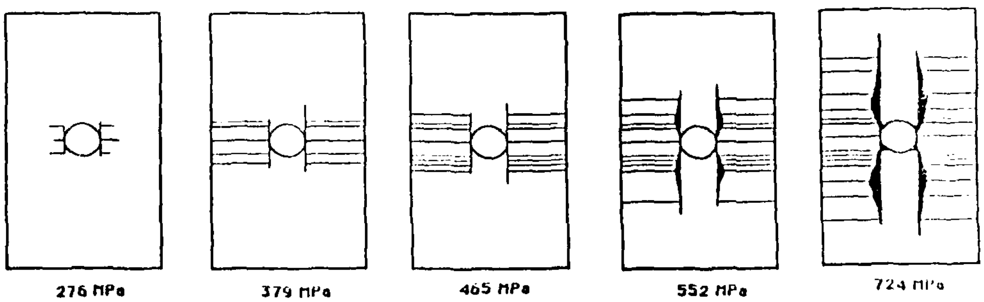


Figure 1. Progression of Matrix Cracking Under Ascending Laminate Stress

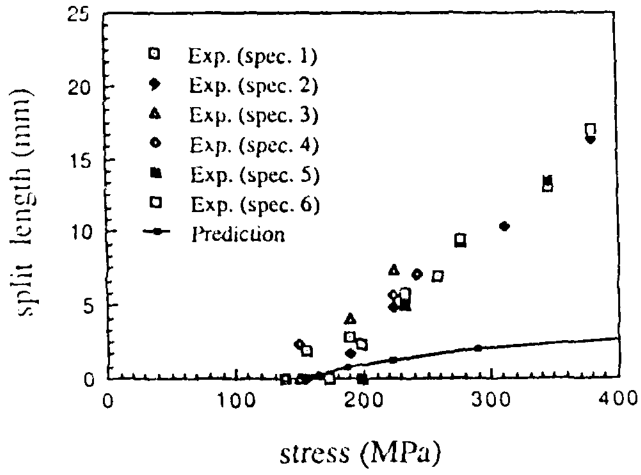


Figure 2. Splitting Length Versus Applied Laminate Stress.

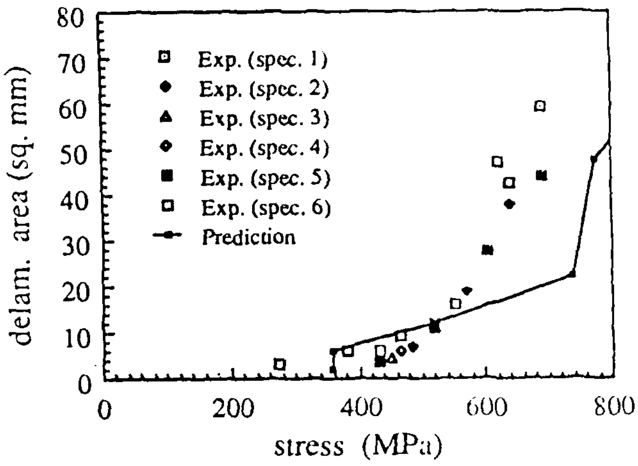


Figure 3. Delamination Area Versus Applied Laminate Stress

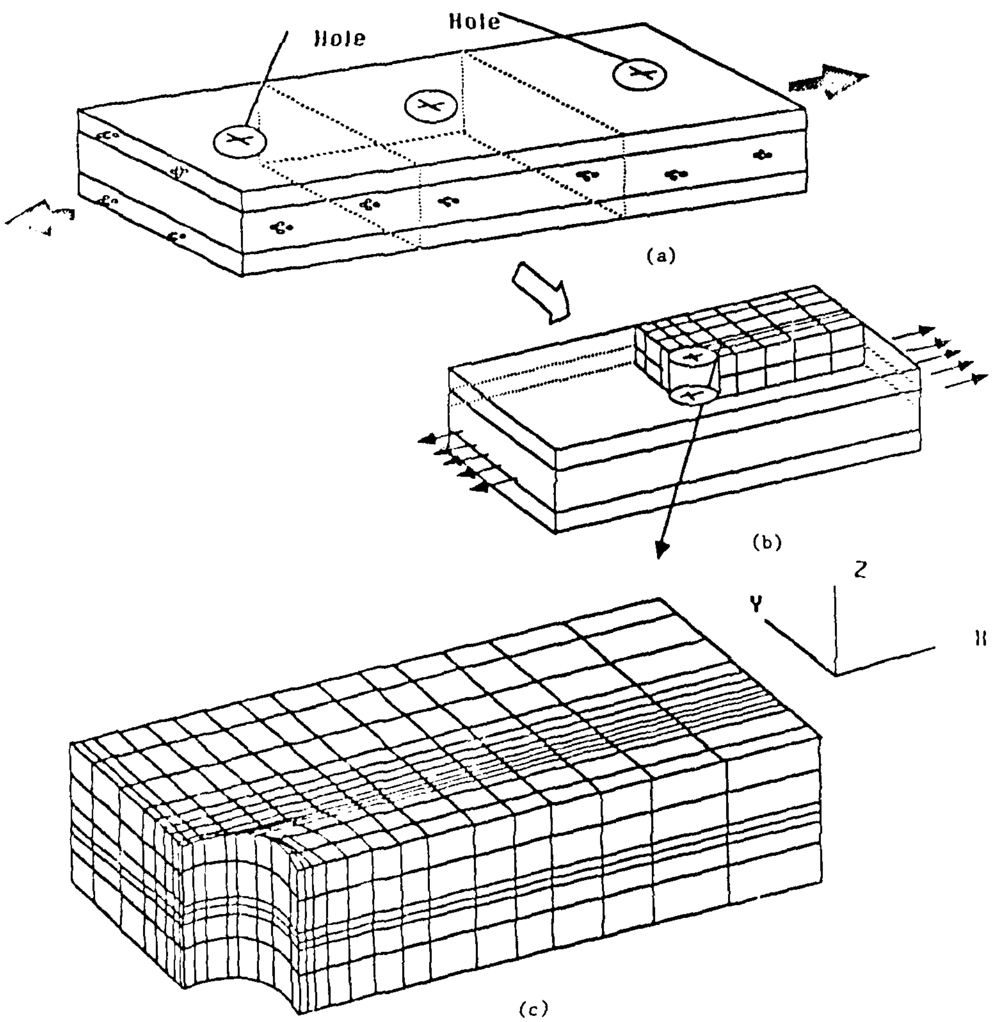


Figure 4. Finite Element Model for the Specimen Containing A Small Hole

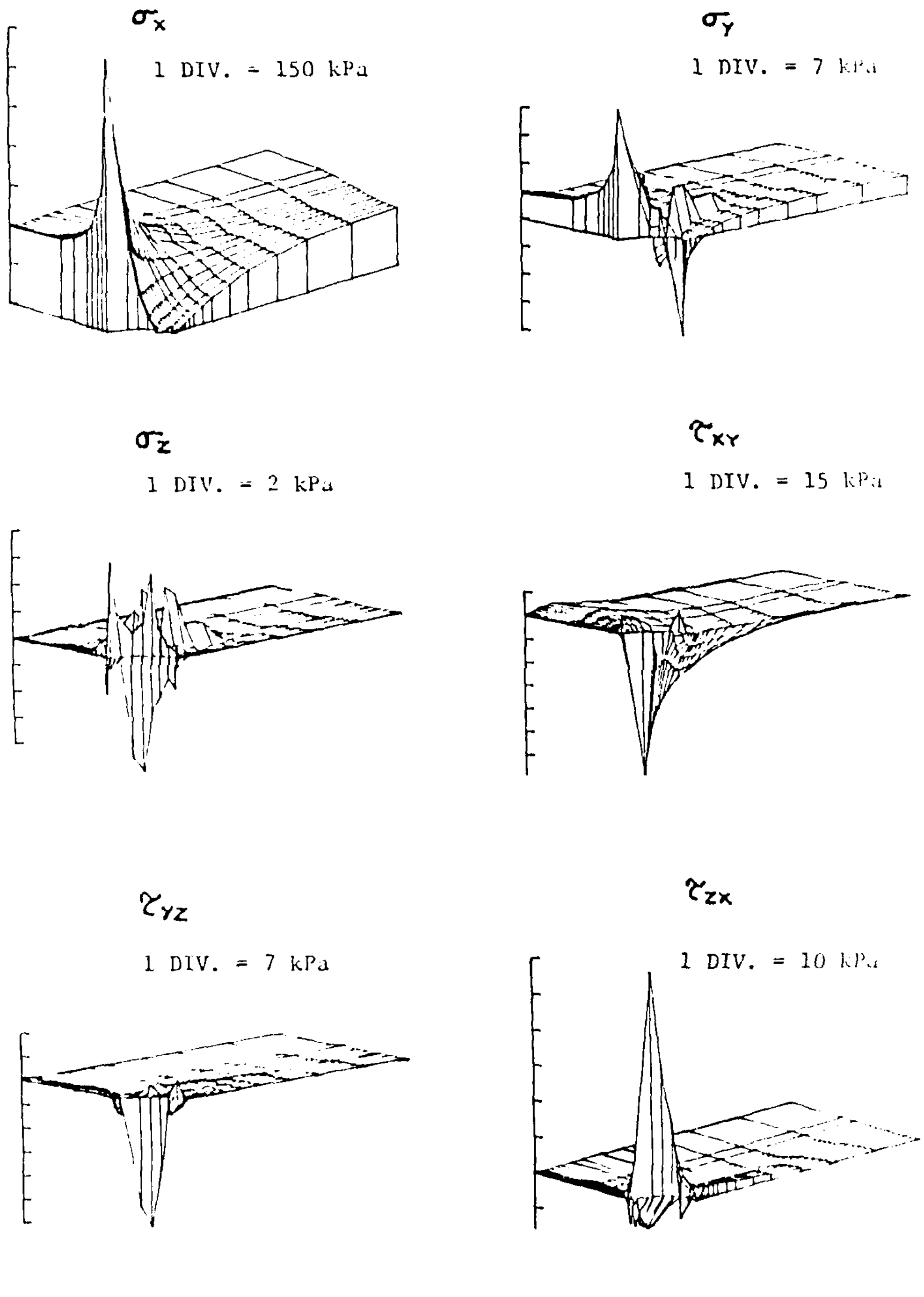


Figure 5. Distribution of Stresses in 0°-Layer Near 0/90 Interface.

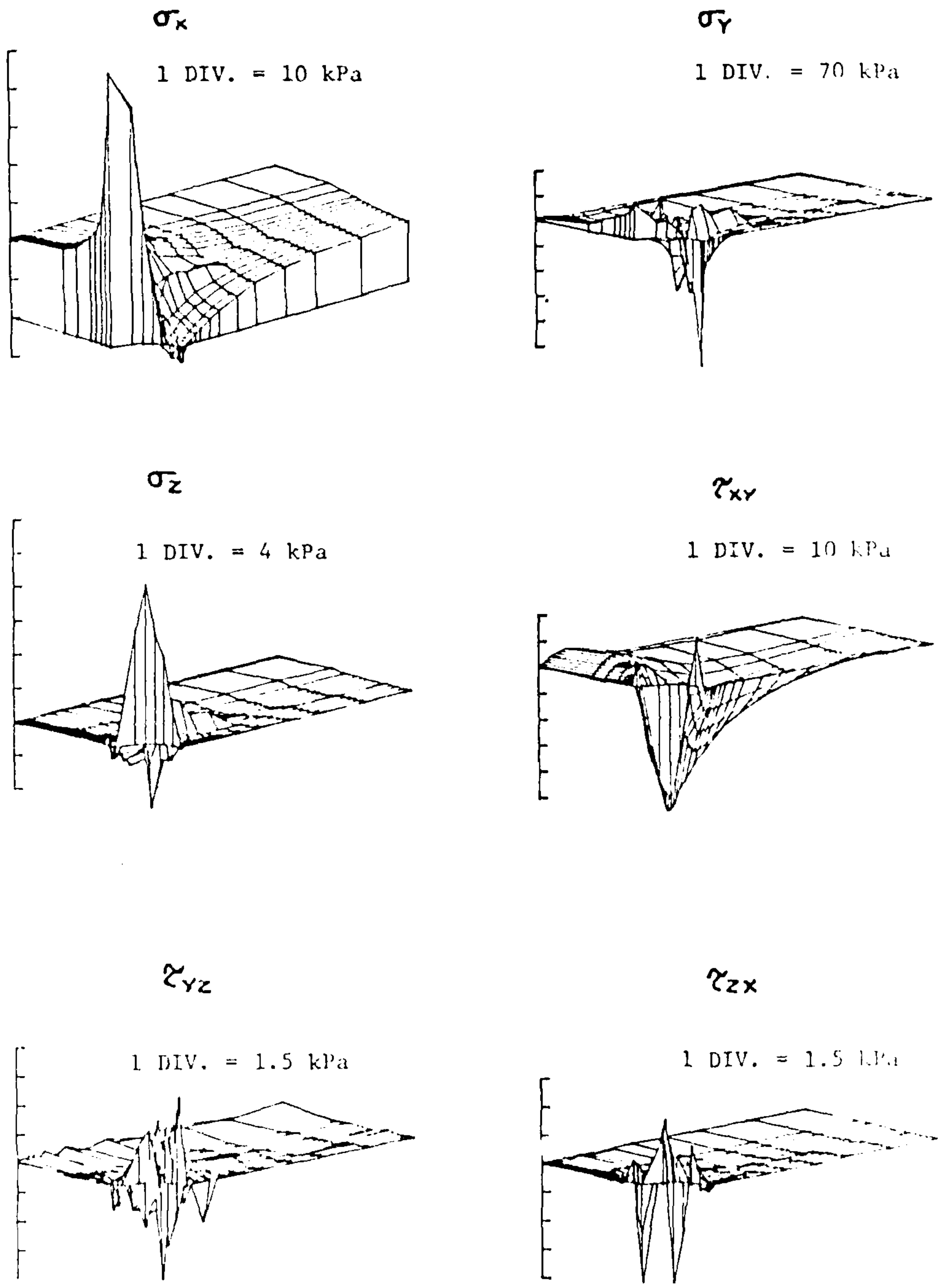


Figure 6. Distribution of Stresses in 90°-Layer Near 90/90 Interface.

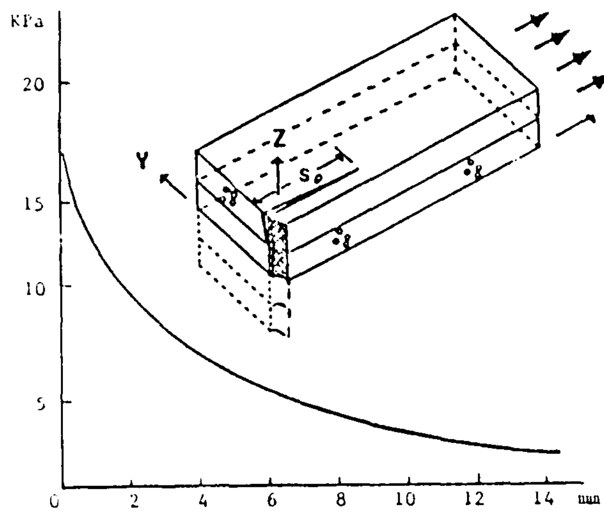


Figure 7. Split-Tip Normal (tensile) Stress Versus Split Length.

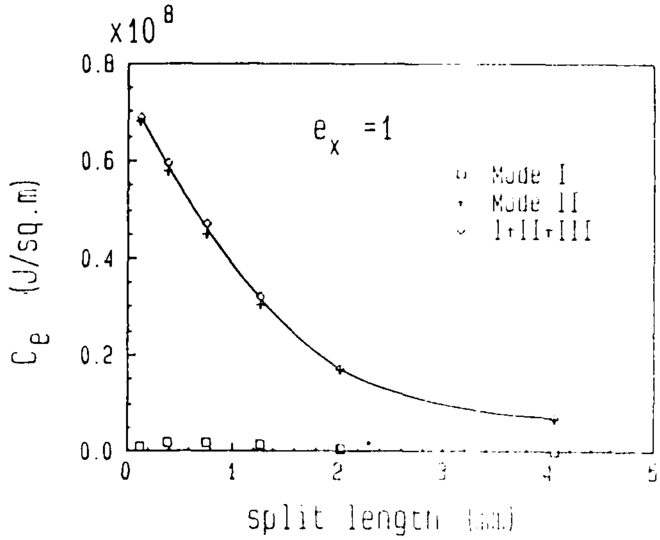


Figure 8. Energy Release Rate Coefficient C_e for 0°-Layer Splitting.

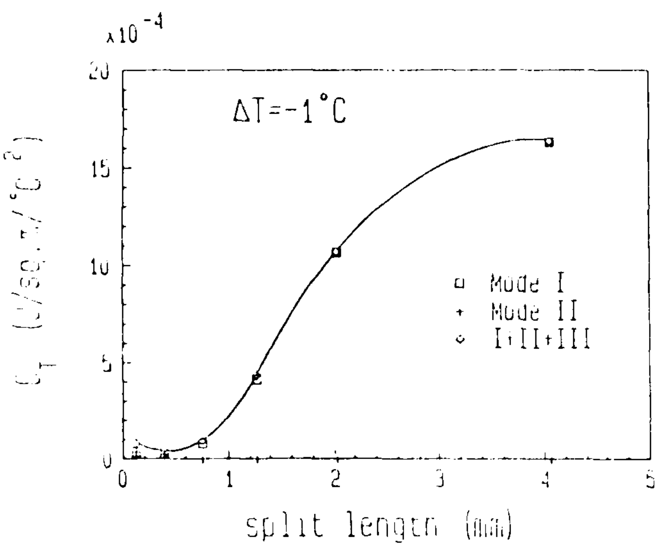


Figure 9. Energy Release Rate Coefficient C_T for 0°-Layer Splitting.

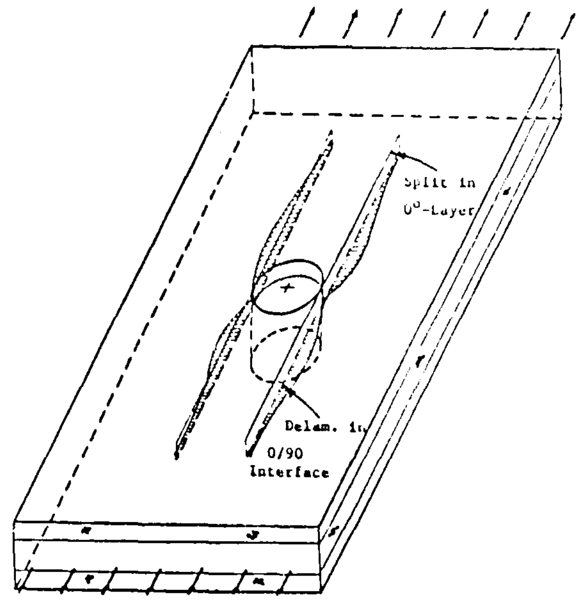


Figure 10. Simulation Model for 0/90 Interface Delamination.

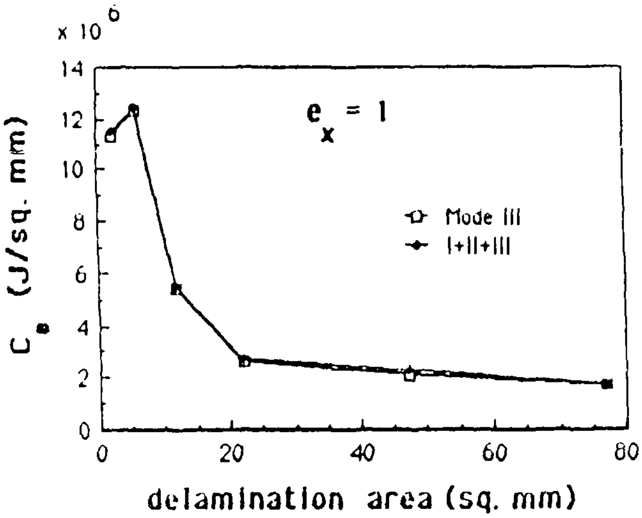


Figure 11. Energy Release Rate Coefficient C_e for 0/90 Interface Delamination.

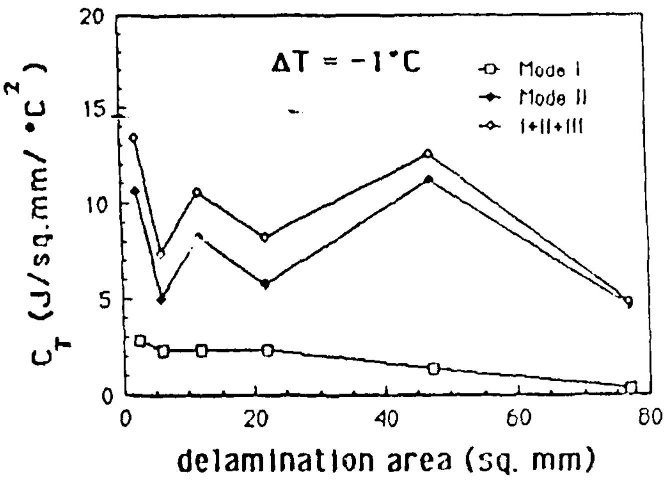


Figure 12. Energy Release Rate Coefficient C_T for 0/90 Interface Delamination.

A COMPREHENSIVE STUDY

ON DAMAGE TOLERANCE PROPERTIES OF

NOTCHED COMPOSITE LAMINATES

Appendix V

3-D Finite Element Crack Simulation Code

User's Guide and Source Code

CONTENTS

1 GENERAL PROGRAM CHARACTERISTICS

1.1 INTRODUCTION

This computer code has been developed for an independent and self contained operation. The program is written in FORTRAN 77 language, adoptable to any medium or large computer. The main function of the program is to simulate numerically the initiation and growth of a plane crack(s) in a 3-D solid, specifically, delamination or splitting or delamination with a split in composite plates. The plate may be subjected to either mechanical loading, thermal loading or both. In order to determine the layer interface which is likely to suffer delamination under the given loading, a search must be conducted by computing the interlaminar stresses. Once the site of delamination is determined, the program will then simulate the delamination growth under the applied loads.

The present computer code can handle (i) splitting along the fiber directiona, (ii) delamination having a plane-contour of arbitrary shape and (iii) delamination in the presence of an opened split. The changes in the boundary conditions as the delamination grows are automatically adjusted in the program. There is no limitation to the number of layers or the stacking sequence. The layers may have different thicknesses and material properties. Each layer is assumed to be a homogeneous, orthotropic elastic medium with one of its principal axes aligned in the thickness directions of the plate (z-axis).

The code is divided into three independent programs: the preprocessor, the main code, and the post processor. The separation of the code in three stages allows modifications to be made in the data at the end

of each particular program so that certain parametric studies can be performed in one stage without repeating the calculations performed in the previous stage.

1.2 THE PREPROCESSOR PROGRAM

This is the first stage in the solution of the delamination problem. The input data necessary for this program consists of the specimen geometry, mesh plan, layer material properties, boundary conditions and the double nodes (double nodes are a pair of nodal points which occupy the same spatial position). The output of this program consists of the full details of the finite element mesh together with the numbered nodes, including the double nodes. Although this output data is sufficient to run the second stage, the data to be input into the main code, the data still needs to be supplemented with the crack opening sequence data set which can be formulated only following the output from the preprocessor.

1.3 THE MAIN CODE, KSAP II

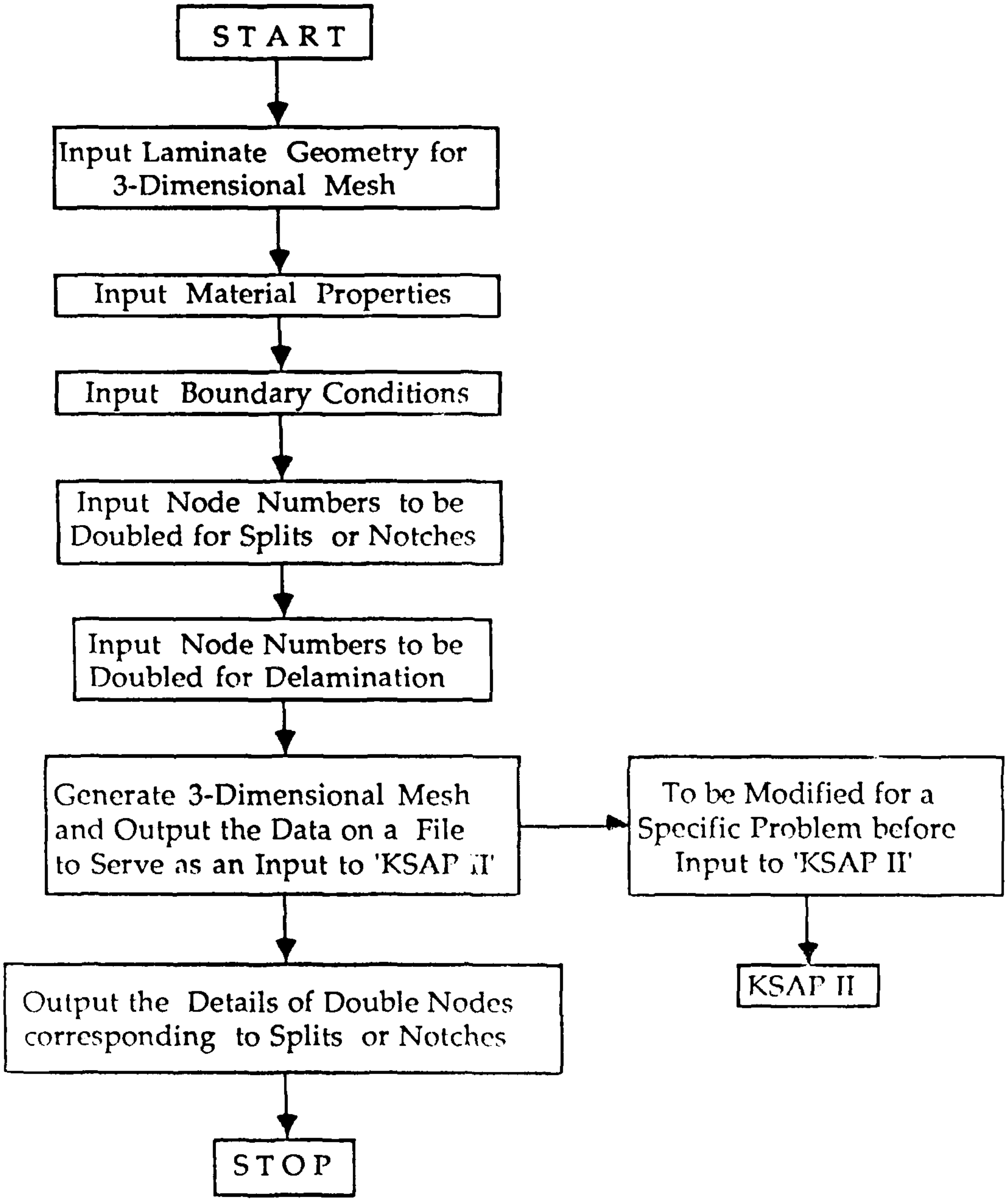
As the name implies, this is the main part in the solution procedure. The output data from the first stage, together with the crack opening sequence data serves as the input data for this program. The program solves the three dimensional problem using an 8 or 21 node solid element with three degrees of freedom (x,y,z) for each node.

1.4 THE POSTPROCESSOR PROGRAM

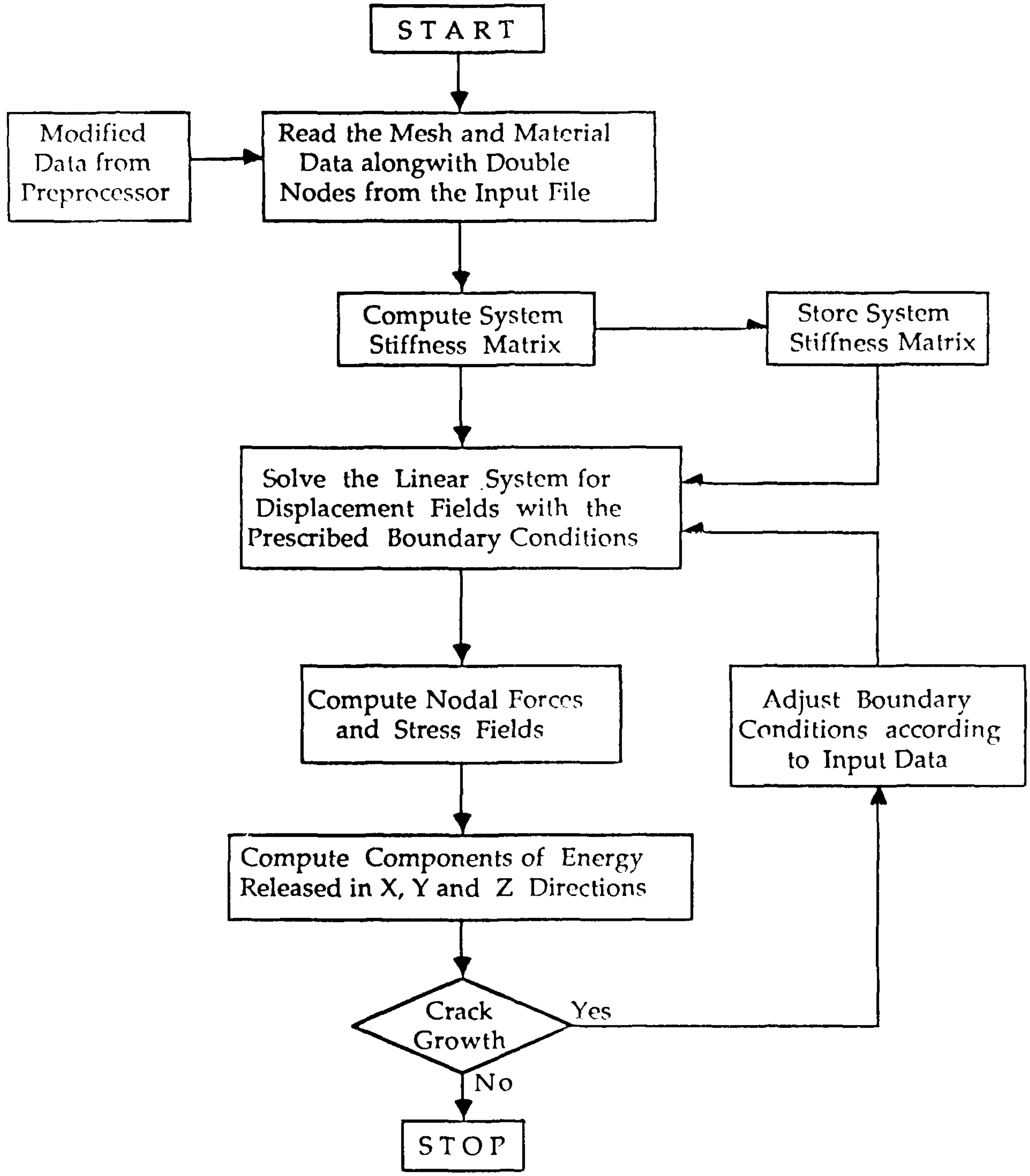
The post processor mainly produces 3-D plots of the stresses with hidden lines removed. The input data for this programs is a modified

output file from the KSAP II program. Various stress distribution plots can be output along any specified plane. The three-dimensional plots can be processed at any specified viewing direction.

The details of preprocessor program and the input can be found in Chapter 2. The modifications to the preprocessor output which are needed before it can be used further are found in Chapter 3. Chapter 4 describes the features of the KSAP II code and Chapter 5 describes the details of postprocessor program. Chapter 6 contains an illustrative example to explain the working of the total code. The FORTRAN source listings of the various parts of the code and their outputs are included in the Appendices A through D.



FLOW CHART FOR 'PREPROCESSOR' PROGRAM



FLOW CHART FOR 'KSAP II' PROGRAM

2 THE STRUCTURE OF THE PREPROCESSOR PROGRAM

2.1 INTRODUCTION

The preprocessor program generates the input data required for the main code, KSAP II. The input data required for the preprocessor program pertains to the dimensions of the plate, mesh plan, material properties of the layers, and the boundary conditions. In it's present form, this program can generate data only for brick type elements with either 8 nodes representing the 8 corners of the element or 21 nodes as shown in Figure 2.1.

There are two options in generating the mesh, one is for rectangular mesh for laminates without any curved boundaries and the other is for generating mesh in a laminate with a central hole. The mesh pattern in the later one is chosen to accomodate split(or split growth) tangential to the hole boundary along the loading direction. There is no limitation on the number of layers or the stacking sequence. Depending on the symmetry in geometry and/or loading, one-half, one-quarter or one-eighth of the plate may be analyzed. The displacement and force boundary conditions have to be appropriately specified in order to take the advantage of symmetry.

The program automatically assigns numbers to nodal points, and cartesian coordinates to each node according to input data. The nodes are numbered in an orderly fashion in x, y, z-directions and the 8 (or 21) nodes for each element can be generated arbitrarily from the set of coordinates given in x, y and z directions. The dimensions of elements in any direction can be controlled by changing the coordinates in that direction and thereby the density of the mesh in any region can be changed.

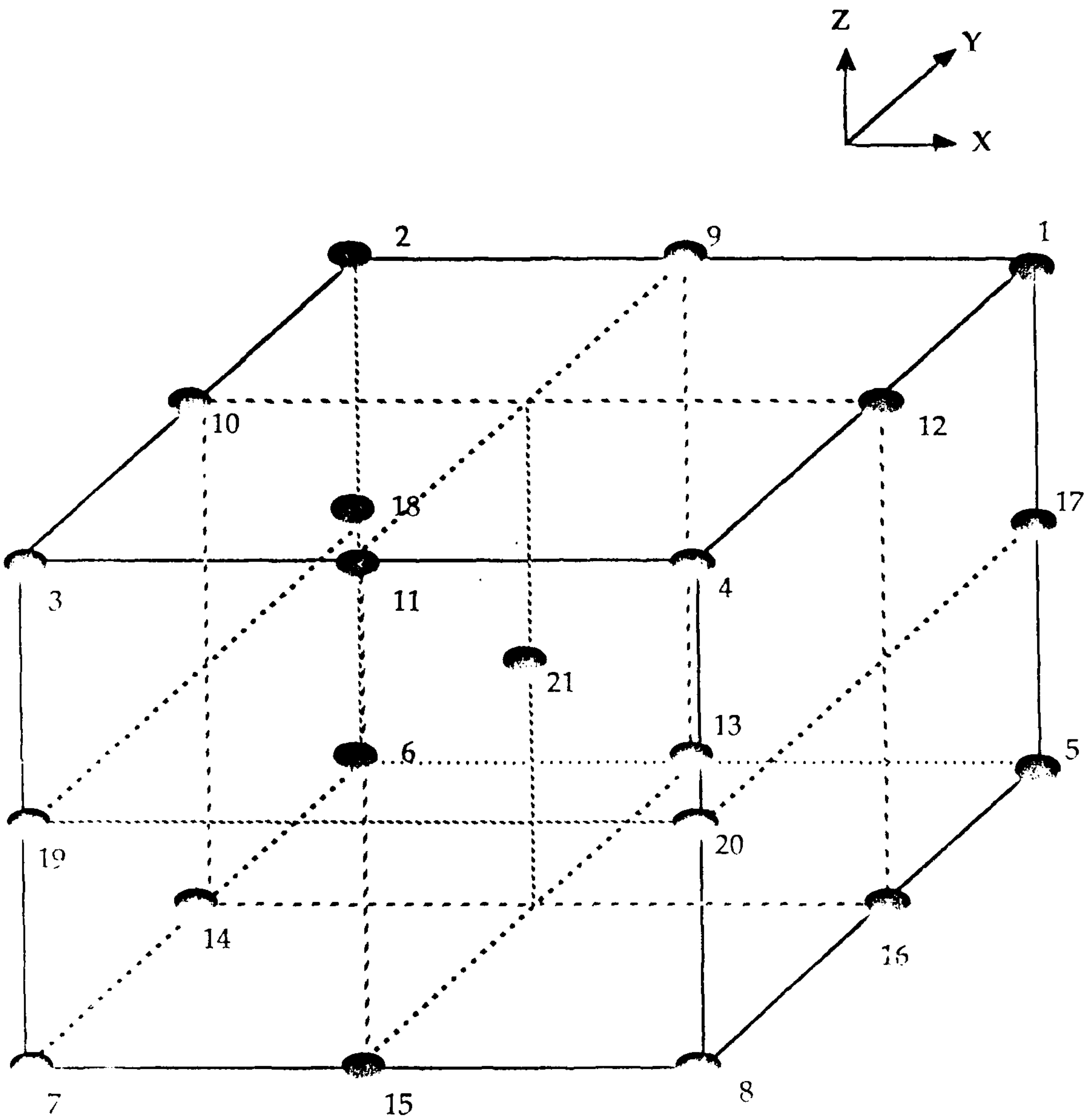


FIG. 2.1 THREE DIMENSIONAL 21 NODE ISOPARAMETRIC ELEMENT

The thermal loading simulatation requires two data sets: assigning stress free temperature for each element and prescribing the temperature at which the plate is to be analyzed for delamination. The stress free temperature is assigned to each element while generating the elements. The temperature at which the plate is to be analyzed is provided while generating nodal points. The temperature distribution need not be uniform for the whole plate and each node can be assigned a different temperature. The details of this data input is explained in the next section 2.2.

Mechanical loading can be either a prescription of forces or a prescription of non-zero displacements at the nodes. The details of prescribing force boundary conditions are found in section 2.2. A plate subjected to uniform strain can be simulated by assigning non-zero displacements to the appropriate nodes. These non-zero displacements are changed to force boundary conditions by attaching a linear spring with a large stiffness value (k) to each node in the given displacement (d) direction and applying a force (=k x d) at the other end of the spring. These boundary elements do not increase the total degrees of freedom of the stiffness matrix. The nodes having zero displacements, which are used to specify the symmetric planes, do not make use of these boundary elements and they essenetially remove those degrees of freedom from the system of equations.

The main program, KSAP II can simulate a crack opening along a symmetric plane or along any plane given by x=constant or y=constant or z=constant. For example: the interlaminar boundary (layer interface) corresponds to z=constant. A crack along a symmetric plane (e.g. the mid-plane of a symmetric laminate) is simulated by suitably changing the boundary conditions at those nodes on that plane, which will be released to

simulate crack opening. The degrees of freedom of these nodes must be retained if a crack is to be simulated along the plane of symmetry. Hence they should not be removed by giving zero displacement in the direction of crack opening. The crack opening instruction along these nodes is explained in the next chapter. If a crack opening along a plane other than the symmetric plane is to be simulated then double nodes are to be assigned for each nodal point located on that plane. The double nodes need not be taken into account while generating the initial set of elements and nodes. Given the plane of crack (1 for yz-plane, 2 for zx-plane and 3 for xy-plane), the preprocessor program has the capability to renumber the mesh and update the node numbers for each element when the instruction pertaining to the double nodes is supplied.

A complete listing of the preprocessor program can be found in Appendix A. The following flow chart shown on the next page illustrates the general structure of the preprocessor program.

2.2 DATA INPUT TO PREPROCESSOR PROGRAM

The input data required for the preprocessor is made very simple and is kept to a minimum. For example, the element generation (assigning node numbers to the elements) can be done in only a few cards as explained in Section 2.2.2.

2.2.1 Details of the Data Input

The input data is arranged in the following nine groups of cards. Each group consists of one or more cards. Data in the groups I, VI, VII and VIII must be given in the specified format. Each entry must be made in the specified columns and a brief explanation of the entry can be found in entry description. The name listed under 'variable' is the name used for that entry in the program listing. The data in the groups II, III, IV and V may be given in free format. In these groups, if the data does not fit on one card, it may be continued on an immediately following card. Each of the groups II, VI and VIII may have several cards and the program recognizes the termination of that group only when it encounters a card with -1 as the first entry.

Group I Heading Card (Format A72):
HED(72) - heading information to be printed with the outputs

Group II Mesh Generation Cards (Free Format)
card 1:
NTYPE - Type of Element (8 node or 21 node)
card 2:
NONX,NONY,NONZ,RAD
- Number of Nodes in X, Y and Z directions, Radius of the hole (if RAD=0 given, rectangular mesh will be generated.
NOTE: If NTYPE=21, NONX, NONY, NONZ have to be odd numbers.

card 3:
XX(1), XX(2), ...XX(NONX)
- x- coordinates of the nodes in the x- direction.
card 4:
YY(1), YY(2), ... , YY(NONY)
- y- coordinates of the nodes in the y- direction.

card 5:
ZZ(1), ZZ(2), ... , ZZ(NONZ)
- z- ccodinates of the nodes in the z - direction.

NOTE: For 21-node element generation, even numbered coordinates should be middle points of immediate neighboring points.
i.e., for i=2,4,...
xx(i)=(xx(i-1)+xx(i+1))/2.
yy(i)=(yy(i-1)+yy(i+1))/2.
zz(i)=(zz(i-1)+zz(i+1))/2.
Any mistake in the coordinates of even numbered coordinates will be corrected by the preprocessor.

For a laminate with a hole, x and y coordinates can be given with hole center as the origin. The coordinates of some of the nodes will be transformed and results in a mesh as shown in Figure 2.2.

Group III Element, Nodal Property Definition Cards (Free Format)

card 1: Nodal temperature at which analysis is to be carried out
N, TEMP, NEND, INC

- N - starting node number
- TEMP - magnitude of temperature of the node
- NEND - last node up to which nodes have same temperature
- INC - increment between N and NEND
-1, 0.0, 0, 0 - data termination card

NOTE: This temperature will be different from stress free temperature of the element (see next card) for thermal loading only.

card 2: Element stress free temperature
N, TEMP, NEND, INC

- N - starting element number
- TEMP - stress free temperature of the element (curing temp.)
- NEND - last element number up to which elements have same temperature
- INC - increment between N and NEND
-1, 0.0, 0, 0 - data termination card

card 3: Element material definition
N, MATEL, NEND, INC

- N - starting element number
- MATEL- material identification number (ex:1,2,3..)
- NEND - last element number up to which nodes have same temperature
- INC - increment between N and NEND
-1, 0, 0, 0 - data termination card

NOTE: This card is to identify the elements as to which material they belong to. Material properties for different identification numbers are given in later cards.

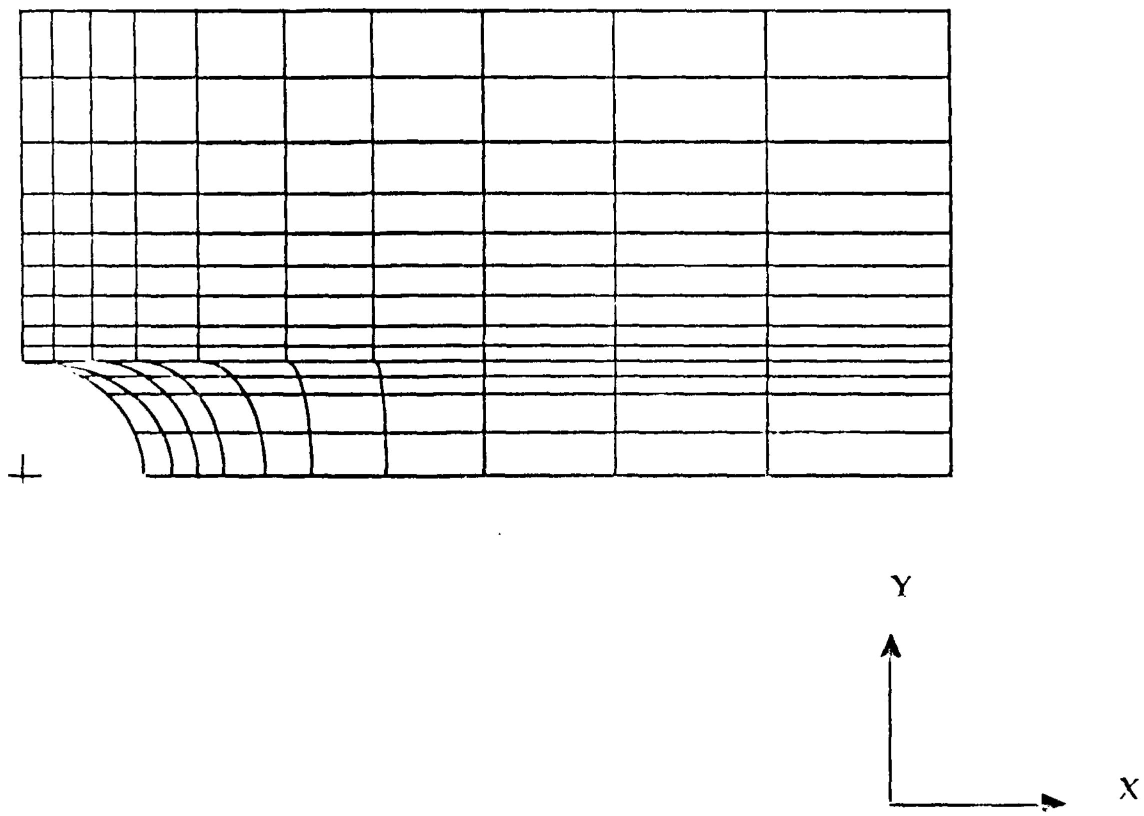


FIG. 2.2 FINITE ELEMENT MESH IN XY-PLANE IN A LAMINATE WITH A HOLE

card 4: Element material axis orientation definition
N, MORTT, NEND, INC

- N - starting element number
- MORTT- material axes orientation set number (ex:1,2,3..)
- NEND - last element number up to which nodes have same temperature
- INC - increment between N and NEND

-1, 0, 0, 0 - data termination card

card 5: Element stiffness matrix reuse definition
M1, M2
M3, M4
\- -
\- -

- M1,M3 - starting element number
- M2,M4 - last element number upto which element stiffness is same

-1, 0 - data termination card

NOTE: These cards to identify the elements with the same stiffness matrix and thereby saves computational time. A number of ranges (M1,M2; M3,M4;..etc.) can be given one after another.

Group IV Split or plane notch definition data (Free Format)

card 1:
NSD,IDIR

-NSD - number of nodes lying inside the split or plane notch region
-IDIR- direction number normal to the plane of the notch
if the normal is parallel to x- axis, IDIR=1
if the normal is parallel to y- axis, IDIR=2
if the normal is parallel to z- axis, IDIR=3

NOTE: If no split is required enter 0,1 and skip card 2.

card 2: node numbers defining split region
N, NEND, INC

- N - starting node number
- NEND - last node number
- INC - increment between N and NEND

-1, 0, 0, - data termination card

NOTE: With this card, number of splits can be defined in parallel planes. Split defined by this card is simulated by doubling the nodes but these double nodes cannot be used to simulate crack growth. In order to read the displacement output of the nodes inside the split, refer to corresponding double nodes given in the output file 'FOR010.DAT'.

Group V   Delamination definition data (Free Format)

card 1:
NTD
-NTD - total number of nodes defining delaminatiom region
If there are no double nodes in the problem enter
0 and skip card 2

card 2: node numbers defining delamination region
NOND(1), NOND(2), ...NOND(NTD)

NOTE: The double nodes and the corresponding original nodes are not written in a separate output file. They are given in KSAPIN.DAT itself. They are arranged in the ascending order of the original nodes to fecilitates easy modification of KSAPIN.DAT for crack growth simulation. So, it is advisable to give the nodes here in the order of their release.

For 21 node element, face center nodes are not used in KSAP II and hence they are eliminated from the double node list by the preprocessor.

card 3:
XL, XU, YL, YU, ZL, ZU

- XL, XU - lower and upper bounds of x- coordinate of the laminate boundary in which second set of double nodes are to be placed.

- YL, YU - lower and upper bounds of y- coordinate of the laminate boundary in which second set of double nodes are to be placed.
- ZL, ZU - lower and upper bounds of z- coordinate of the laminate boundary in which second set of double nodes are to be placed.

Group VI   Material Property Data

Orthotropic, temperature dependent material properties may be prescribed. Here L,T,Z are the principal axes of the material. For each different material the following group (a) cards must be supplied.

(a) Material Properties (Format A4,I4,4X,G17.7)

card 1:

columns	entry description
1- 4	bbEL
5- 8	material identification number
13-29	value of Young's modulus in L-direction

card 2:

columns	entry description
1- 4	bbET

5- 8	material identification number
13-29	value of Young's modulus in T-direction

card 3:

columns	entry description
1- 4	bbEZ
5- 8	material identification number
13-29	value of Young's modulus in Z-direction

card 4:

columns	entry description
1- 4	NULT
5- 8	material identification number
13-29	value of the poisson's ratio, ν_{1t}

card 5:

columns	entry description
1- 4	NULZ
5- 8	material identification number
13-29	value of the poisson's ratio, ν_{1z}

card 6:

columns	entry description
1- 4	NUTZ
5- 8	material identification number
13-29	value of the poisson's ratio, ν_{tz}

card 7:

columns	entry description
1- 4	bGLT
5- 8	material identification number
13-29	value of the shear modulus, G_{1t}

card 8:

columns	entry description
1- 4	bGLZ
5- 8	material identification number
13-29	value of the shear modulus, G_{1z}

card 9:

columns	entry description
1- 4	bGTZ
5- 8	material identification number
13-29	value of the shear modulus, G_{tz}

card 10:

columns	entry description
1- 4	ALFL
5- 8	material identification number
13-29	value of the thermal expansion coeff., α_1

card 11:

columns	entry description
1- 4	ALFT
5- 8	material identification number

13-29 value of the thermal expansion coeff., α_t

card 12:

columns	entry description
1- 4	ALFZ
5- 8	material identification number
13-29	value of the thermal expansion coeff., α_z

NOTE: If any of these 12 cards are not supplied then that particular value will be set equal to zero.
The 12 constants (E11,Ett,..., z) are defined with respect to a set of axes (L,T,Z) which are the principal material directions.

(b) Data termination Card (Format A4)

columns	entry description
1-2	-1 indicates the end of material property cards.

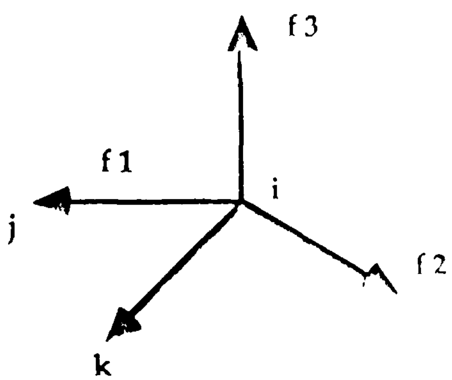
(c) Material Axes Orientation

In this set the data regarding the material principal axes (L,T,Z) relative to the global axes (x,y,z) is furnished. There can be several sets of orientations and one card should be input for each orientation as follows:

columns	variable	entry description
1 - 5	MORT	material axes orientation set number
6 - 10	NI	node number for point "i"
11- 15	NJ	node number for point "j"
16- 20	NK	node number for point "k"

NOTE: Orientation set numbers (MORT) must be input in increasing sequence beginning with "1".

Orthotropic material axes orientations are specified by means of the three node numbers, NI,NJ,NK. For the special case where orthotropic material axes coincide with the global axes (x,y,z), it is not necessary to input data in this section. Let f1, f2, f3 be the three orthogonal vectors which define the axes of material orthotropy then their directions are as shown below:



Node numbers NI,NJ,NK are only used to locate points i,j,k respectively and any convenient nodes may be used.

End the material orientation definition cards with -1 card.

Group VII Force boundary conditions
(Format I6,1X,A4,1X,F10.0,12X,I6,I6)

columns	variable	entry description
1- 6	N	node at which force acts
8-11	LABEL	direction of force (in nodal coodinate system) FX, FY, or FZ
13-22	FORCE	value of the force
35-40	NEND !	If NEND is greater than N (for N positive) all nodes from N thru NEND in steps of INC has this specified force (if INC is left blank it is assumed to be 1)
41-46	INC !	

NOTE: N=-1 signifies the end of this set of cards

Group VIII Displacement Boundary Conditons
(Format I6,1X,A4,1X,F10.0,12X,2I6)

This set of cards is used to constrain nodal displacements to specified values and to compute support reactions. Boundary elements are used to specify strain for the specimen. The boundary element is essentially a spring which has an axial displacement stiffness and it is defined by a single directed axis through the specified nodal point. If any nodal displacement (UX, UY or UZ) is specified to have 0.0 value then that degree of freedom is eliminated from the stiffness matrix.

columns	variable	entry description
1- 6	N	node number at which this displacement will be used
8-11	LABEL	type of displacement boundary condition
13-22	V	value of the displacement
35-40	NEND !	If NEND is greater than N (for N positive) then all nodes N thru NEND in steps of INC will have this specified displacement
41-46	INC !	

NOTE: N=-1 signifies the end of this set of cards.

LABEL can be UX, UY or UZ (upper case) which means that the

specified displacement is in x, y, & z directions respectively.

Group IX Stress Output Locations (Free Format)

LOC1, LOC2, LOC3, LOC4, LOC5, LOC6, LOC7
- location numbers in ascending order

NOTE: In KSAP II, there is a provision to obtain stresses at a maximum of 7 locations in an element. Any 7 of the 27 locations shown in Figure 2.3 can be chosen.

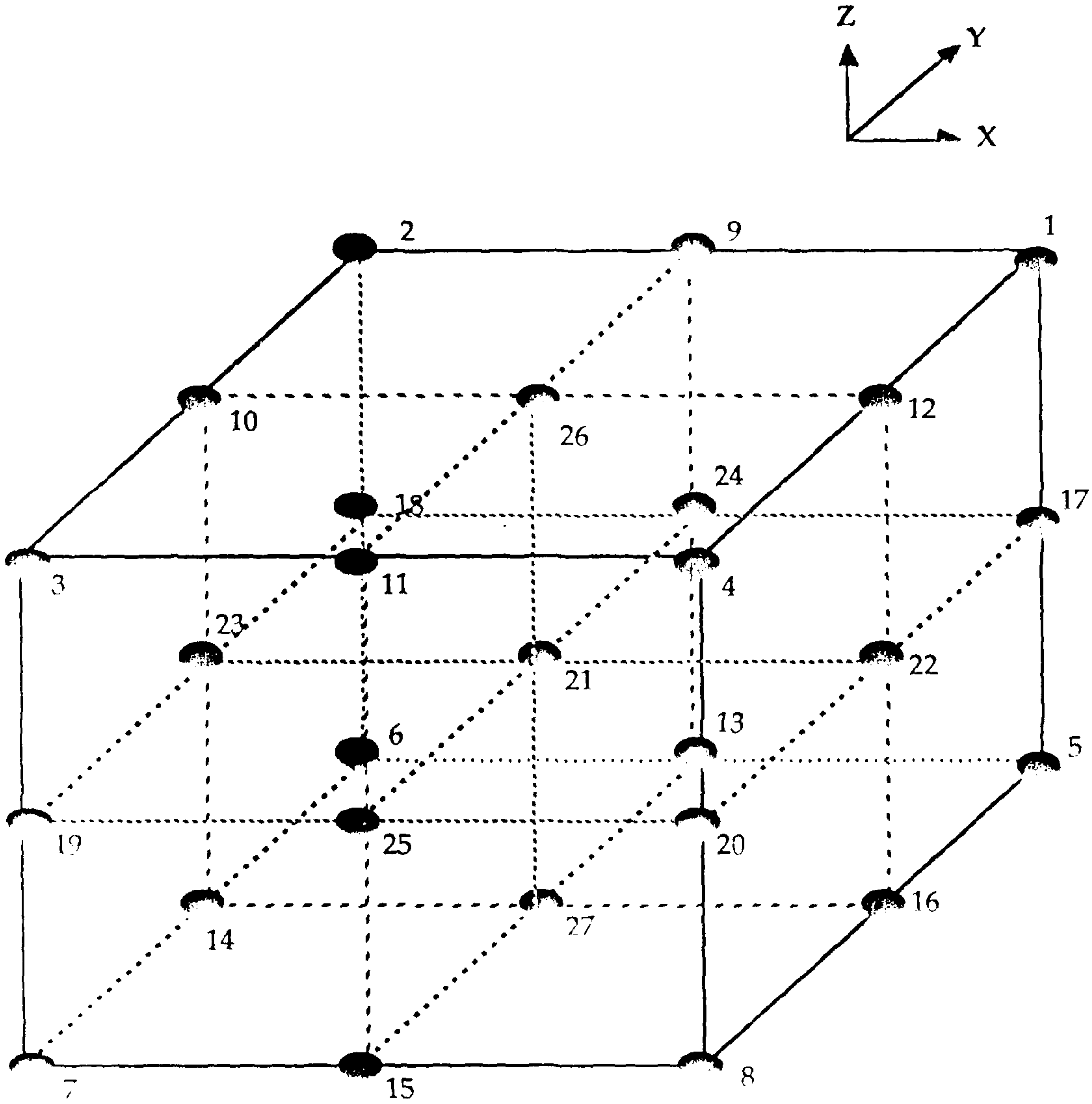


FIG. 2.3 STRESS OUTPUT LOCATION NUMBERS IN AN ELEMENT

3 MODIFICATION OF THE PREPROCESSOR OUTPUT DATA

3.1 INTRODUCTION

The output data of the preprocessor program is to be modified before it can serve as input data to the main code, KSAP II. The preprocessor program will output two files of data: one file will serve as input data file to the KSAP II code and the other file contains the renumbered double nodes for split or notch simulation. The first file is to be supplemented with information regarding the location of double nodes and the information regarding crack opening node sequence. In addition, it is also possible to give commands to selectively print the stress ouput.

3.2 DOUBLE NODES AND CRACK OPENING SEQUENCE

A double node is originally one node which has two node numbers. These are provided in the plane along which the crack propagation is to be simulated. The double nodes serve two purposes. If the displacements of both the nodes are specified to be same, then they behave as one single node. If, on the other hand, the displacements of the nodes are specified to be independent then they behave as two separate nodes thus simulating crack propagation through that node. Usually, a node has three degrees of freedom, in x,y,z-directions. In the case of double nodes each node has three degrees of freedom after they are separated. However, if the double node is on a symmetric plane, then each node will not have three degrees of freedom after they are separated. For example, Figure 3.1 shows the one-eighth part of a laminate subjected to a force in y-direction. x=0, y=0 and z=0 are symmetric planes. Let there be a transverse crack in the bottom ply as shown by the shaded area. The double nodes 50 and 51 are not

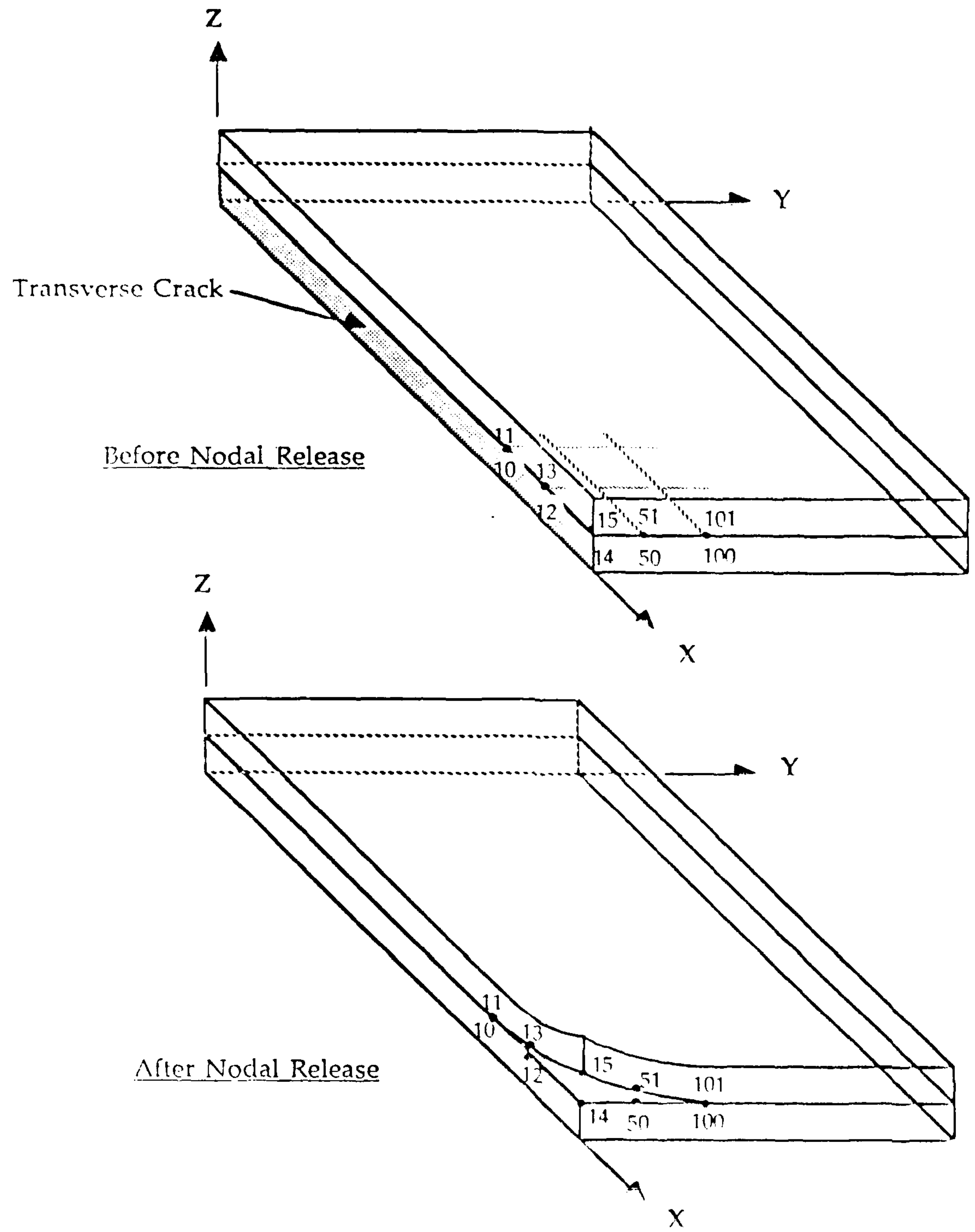


FIG. 3.1 DOUBLE NODE ARRANGEMENT FOR DELAMINATION SIMULATION

on any symmetic plane and hence, the three displacements (Ux,Uy, & Uz) of node 50 are respectively the same as those of node 51 before the nodes are separated. The nodes 50 and 51 each will have three independent degrees of freedom once they are separted. However, the double nodes 10 and 11 will behave differently. When they are together Uy=0 since they lie on the symmetric plane y=0; and Ux and Uz of node 10 are equal to Ux and Uz of node 11. When the two nodes are separated as the crack propagates, the upper node 11 will be still on the symmetric plane and it will have Ux and Uz degrees of freedom, whereas the bottom node 10 is no more constrained and will have all the three dgrees of freedom, Ux,Uy and Uz, free.

3.3 DETAILS OF DATA MODIFICATION

At the end of the preprocessor output the following cards have to be added with regard to double nodes and the crack opening sequence:

I) Details of the constrained degrees of freedom (Free format)

a) NB - total number of degrees of freedom of those double nodes which are constrained by the symmetric plane or constrained by specified displacements before they are released.

b) Details of the constrained degrees of freedom. There should be NB following cards. Each card contains the following input:

NBC - node number
IFIX - degree of freedom
1 for x, 2 for y, 3 for z degree of freedom
DSAVE - value of the specified displacement
(=0, for nodes on that particular symmetric plane)

II) Details of the other double nodes' degrees of freedom

a) NPAIR - number of pairs of all double nodes including those constrained on the symmetric plane

b) There should be NPAIR following cards. Each card will give details of one pair of double nodes and the possible degrees of freedom after the nodes are open.

NPC1 - node number 1 of the pair
NPC2 - node number 2 of the pair
NPX - 0 or 1

NPX=1 signifies that the two nodes are constrained to have the same displacement Ux, before the nodes are open and the nodes are completely free of each other in x- direction of freedom after they are open. NPX=0 signifies that their degrees of freedom are already specified as explained in Group I.

NPY - 0 or 1 ! In Y and Z directions similar to the X-
NPZ - 0 or 1 ! direction as explained above for NPX.

III) Data for each step of opening of nodes:

a) Opening of the double nodes to simulate crack propagation (Free format)

N1 - ! the node numbers of the paired double nodes
N2 - ! which are to be opened
IDF - 1,2 or 3 the degree of freedom which is to be freed. IDF=1 denotes x-degree of freedom is freed from its double node's x-degree of freedom and the nodal force Fx becomes 0. Likewise, IDF=2 or 3 denote y or z-degree of freedom is freed.

For those nodes on the symmetric plane and constrained by by specified displacement which are to be freed the above card should be modified as

N1 - node number
N2 - 0
IDF - 1,2, or 3 depending on x, y, or z degree of freedom which is to be freed

There should be as many cards as there are degrees of freedom to be freed.

N1=N2=IDF= 0 signifies the end of this crack opening instruction and the stress and energy released associated for this opening will be calculated.

b) Selective stress print option (Free Format)

NBEG ! stresses will be output for the element from
NEND ! elements from NBEG thru NEND
This card should not be left blank. If 0,0 is entered stresses will not be printed.

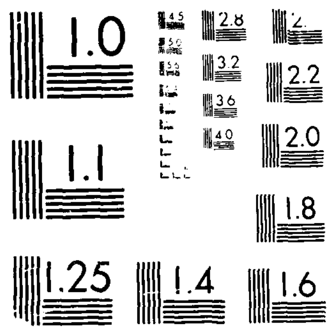
If another step of opening is desired, the above (a) and (b) will be repeated. This may be continued until all the

nodes in Group-I and II are relaxed.

It may be noted that if the displacement and stress solution is desired before any crack is simulated a 0,0,0 card is necessary after Group I and Group II cards.

IV) The crack propagation is terminated by a card containing '9999 9999 0' as input.

AD-A192 952 A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF 2/3
NOTCHED COMPOSITE..(U) DREXEL INST FF TECH PHILADELPHIA
PA DEPT OF MECHANICAL ENGINE.. A S WANG ET AL. FEB 88
UNCLASSIFIED AFOSR-TR-88-0280 AFOSR-84-0334 F/G 11/4 NL



MICROCOPY RESOLUTION TEST CHART

4 THE STRUCTURE OF THE MAIN CODE, KSAP II

4.1 INTRODUCTION

KSAP II is the main program in the analysis of a delamination or splitting problem of a composite plate. The program simulates the crack opening using the data regarding the finite element mesh and the predetermined crack opening sequence. At each step, the program computes the energy released together with the stress and displacement fields.

4.2 GENERAL FEATURES OF THE CODE

The code uses an 8-node or 21-node solid brick element to calculate the stiffness matrix. Each node is assumed to have three degrees of freedom in x, y, z-directions. General orthotropic material properties can be assigned to the element. It is assumed that the whole element is at a uniform temperature given by the average of the temeperatures at the 8 (or 21) nodes. The thermal loads are calculated using the difference between this average temperature and the stress free temperature of the element.

KSAP II code has the capability to simulate crack opening along the surface which passes through the points where double nodes are prescribed. Initially, the two nodes in each pair are assigned the same displacements in the three degrees of freedom. The system of linear equations are solved with the appropriate boundary conditions (mechanical loading or thermal loading or both) for nodal displacements and nodal forces. The stresses at the prescribed locations in each element are also calculated. The nodal forces of the double nodes are nothing but the internal forces holding these two nodes together. These forces are stored and will be used in the

next iteration as the crack opening is simulated through those nodes. The crack opening is simulated by changing the boundary conditions of the double nodes. This implies, obviously, that the displacements of the two nodes will not be the same. Then the system of linear equations are solved for nodal displacements and nodal forces under the changed boundary conditions. The difference in the displacements of the two nodes through which crack opening is simulated will be the crack opening displacement. Using the internal force which was necessay to hold them together (as found in the preceding iteration), the strain energy released can be computed as the crack opening is simulated through that node. This procedure can be continued until all the double nodes are opened.

Thus, strain energy released as the crack passes through successive double nodes can be calculated at each step. At each step the crack opening can be simulated through one or more pairs of double nodes and there is no limitation on the crack front shape.

If the crack is simulated along a symmetric plane, there is no need for double nodes in that plane. The crack opening can be done by simply changing the boundary conditions of the nodes on that plane from displacement boundary conditions to free force boundary conditions.

Once the strain energies released are calculated at each step then the energy release rates (energy resleased per unit area) may be obtained by dividing the energy released by the increment in crack area at that step.

A complete listing of the KSAP II code can be found in Appendix B. The following flow chart illustrates the general structure of the KSAP II program.

4.3 THE OUTPUT DATA FROM KSAP II

The output data from KSAP II consists of the details of the finite element mesh of the given problem as well as the solution of the laminated plate for the given crack simulation. For easy reference, the stresses and energies released are written in separate files, KSAPOUT.DAT and WORK.WOK respectively. The rest of the output (control information and displacements) is written in DISP.OUT.

The output file DISP.OUT contains the following information:

i) control information

ii) the nodal point data: cartesian coordinates of each node, the temperature at each node, and the boundary condition codes (1 means restrained, 0 means free to move in that degree of freedom)

iii) the equation numbers assigned to each nodal degree of freedom

iv) boundary elements data which are attached to nodes where non-zero displacement boundary conditions are prescribed

v) the material property tables for different layers

vi) element data which consists of corner nodes and the material table number to which the element belongs

vii) data regarding equations, i.e., number of equations, bandwidth etc.

viii) the solution data at each step consists of the nodal displacemets (U, V, W) and forces (Fx, Fy, Fz)

(the first step is numbered as zero, the second step is numbered 1 and so on)

The output file KSAPOUT.DAT contains the following informaion:

i) element stresses (SIG-XX, SIG-YY, SIG-ZZ, SIG-XY, SIG-YZ, SIG-ZX) at the prescribed locations in each element.

ii) from the second step onwards, energy released in x, y, and z directions will also be output after the element stresses. This energy released output is the sum of the

energies released at all the double nodes relaxed at that iteration if more than one double node are relaxed.

4.4 LIMITATIONS OF KSAP II CODE

At present the program can handle upto 3000 nodes, and 100 pairs of double nodes. Should a problem involve finer mesh and more than 3000 nodes then the following dimension statements have to be suitably changed:

1) The degrees of freedon (3 times the total number of nodes) have to be changed in ICR(*), R(*) in the statement with serial numbers 3380, 3386, 3844 and 3849.

2) The number of nodes in ID(*,6) have to be changed in the statementswith serial number 3379.

3) The double nodes' total degrees of freedom (6 times the pairs of double nodes) have to be changed in TMATM(*,*), TMAT(*,*), TCOL(*,1), TCOL(*), TCOLM(*), IST(*) in the statements with serial numbers 3384, 3385, 3386, 3848 and A(*,*), B(*,1), IPIVOT (*), INDEX(*,*), DT(*) in the statement with serial number 4366 to the same value.

5 POSTPROCESSOR PROGRAM

5.1 INTRODUCTION

The postprocessor program 'plot' is designed to present the stress output of the main code, KSAP II, in a graphical form. The stress distribution in any plane parallel to xy plane can be displayed on a graphics terminal or graphics output can be obtained on printronix printer or Hewlett Packard plotter. The code uses 3-D graphics routines from Template package. The program is written to run interactively and the interactive input consists of choice of device, stress number (1 for xx, 2 for yy, 3 for zz, 4 for xy, 5 for yz and 6 for zx stress), viewing position coordinates, scale factor to scale stress values. The stress values, coordinates and the related data are read from a prescribed data file.

5.2 DETAILS OF DATA FILE

For an 8 node element the following data precede stress data:

Heading - data file identification heading

NX, NY, NL

NX - number of coordinates in x direction
NY - number of coordinates in y direction
NL - number of layers of finite elements in the laminate

XX(I), I=1,NX - coordinate values in x direction
YY(I), I=1,NY - coordinate values in y direction

For 21 node element the following data should precede stress data:

Heading - data file identification heading

NNODES, NLOC

NNODES - number of nodes of finite element (8 or 21)
NLOC - number of stress output locations requested

NONX, NONY, NONZ

NONX - number of coordinates in x direction
NONY - number of coordinates in y direction
NONZ - number of coordinates in z direction

LOC(I), I=1, NLOC - stress output locations

X(I), I=1, NONX - coordinate values in x direction
Y(I), I=1, NONY - coordinate values in y direction
Z(I), I=1, NONZ - coordinate values in z direction

At the end of the above set of data, one set of stress output (for all the elements) should be copied from KSAPOUT.DAT.

The following plots (Figs.5.1 and 5.2) are obtained from stress output at step 0 in the example problem. Fig. 5.1 is the normal stress along y direction (stress no. 2 and finite element layer no. 3) in 0° layer. The interlaminar stress (stress no. 3 and finite element layer no. 3) in 0° layer is shown in Fig. 5.2. Both the plots are generated on Hewlett Packard plotter using eye coordinates (30, -30, 30).

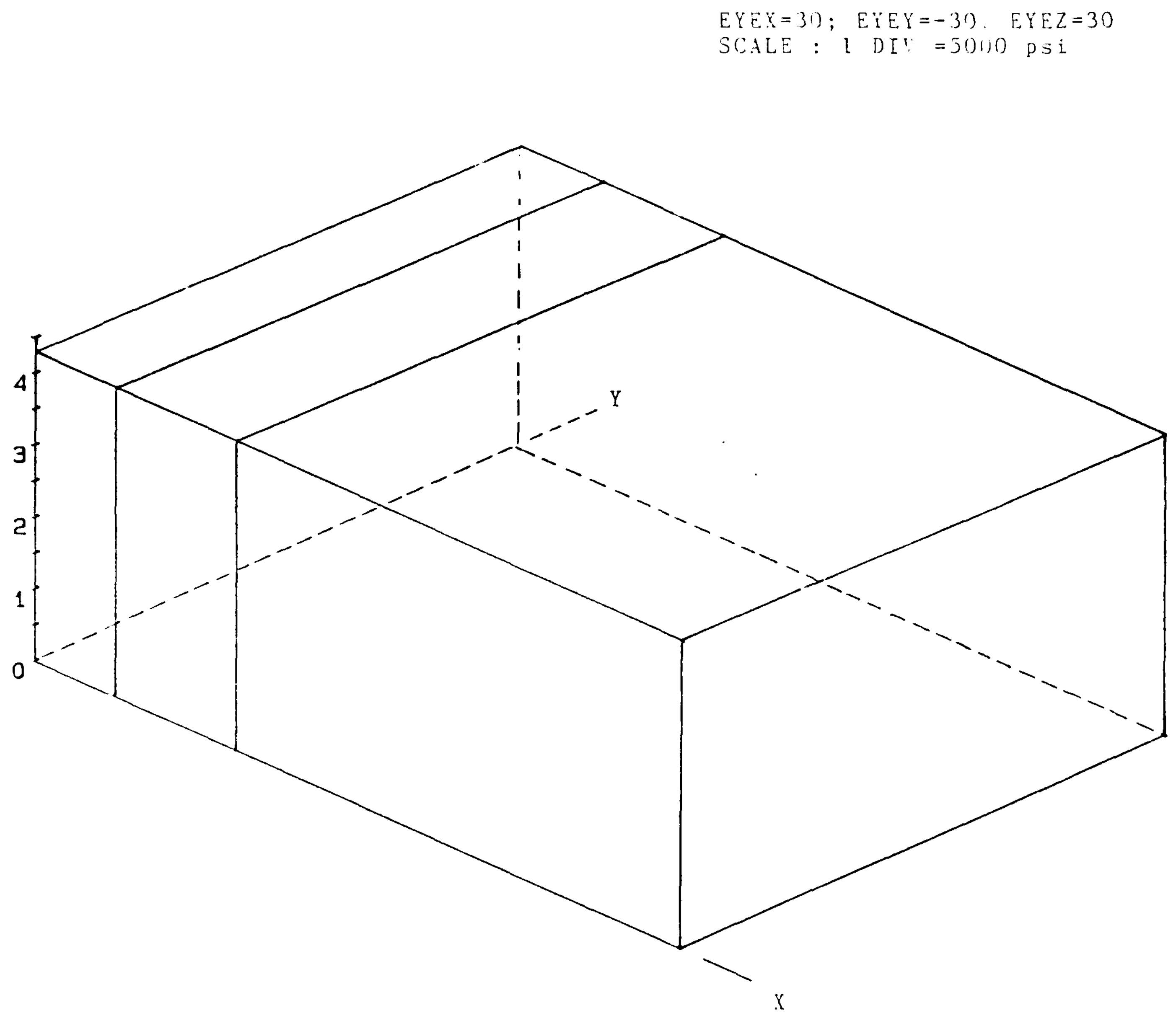


FIG. 5.1 NORMAL STRESS DISTRIBUTION ALONG Y DIRECTION IN 0° LAYER

EYEX=30; EYEY=-3-; EYEZ=30
SCALE : 1 DIV. = 5 psi

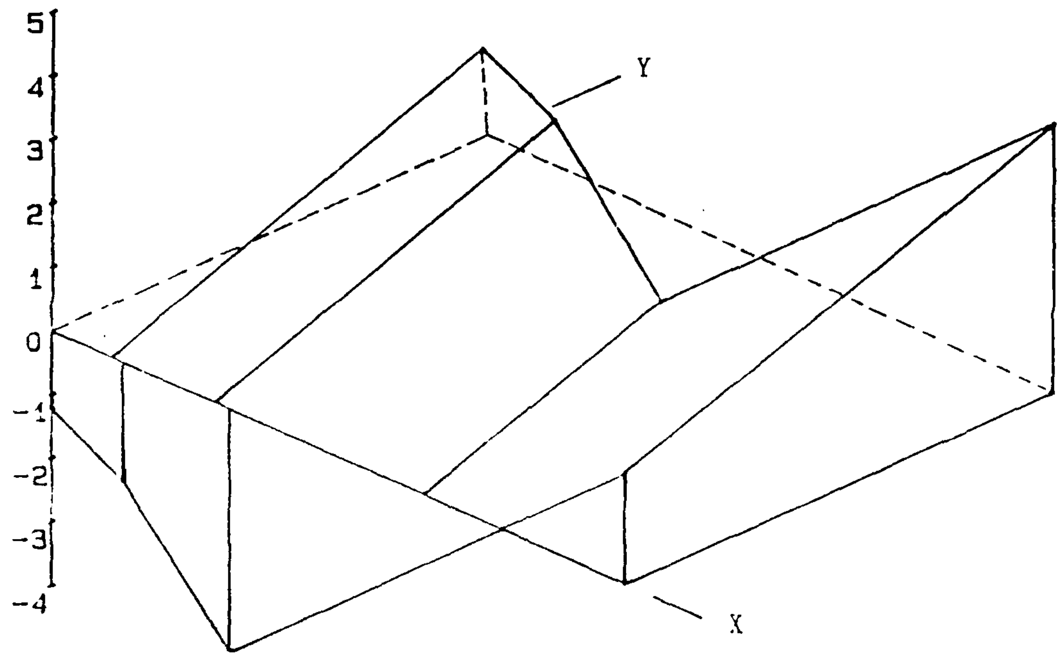


FIG. 5.2 INTERLAMINAR NORMAL STRESS DISTRIBUTION IN 0° LAYER

6 ILLUSTRATIVE EXAMPLE

6.1 INTRODUCTION

In this section we consider an example problem. The example is of a simple laminate construction and it does not represent a practical problem. The purpose here is to illustrate the procedure to operate the computer code. However, the code is developed for a more general use, subjected to the limitations discussed in the preceding sections.

The following paragraphs present the actual working steps in using the present code to generate the interlaminar stress distribution in a given interface plane. All input and output data for this example problem are found in Appendix D.

<u>Laminate Geometry</u>:
(Because of symmetry only one-eighth of the laminate is considered)

width of the laminate is 8.0"
length of the laminate is 6.0"
number of layers is 2
thickness of layer 1 is 1.0"
thickness of layer 2 is 1.0"

As shown in the Figure 6.1 x=0, y=0 and z=0 are symmetric surfaces and a uniform farfield strain is applied in y-direction.

The delamination cracking and mesh size are selected as follows:

4 equal divisions in x-direction
2 equal divisions in y-direction
4 equal divisions in z-direction

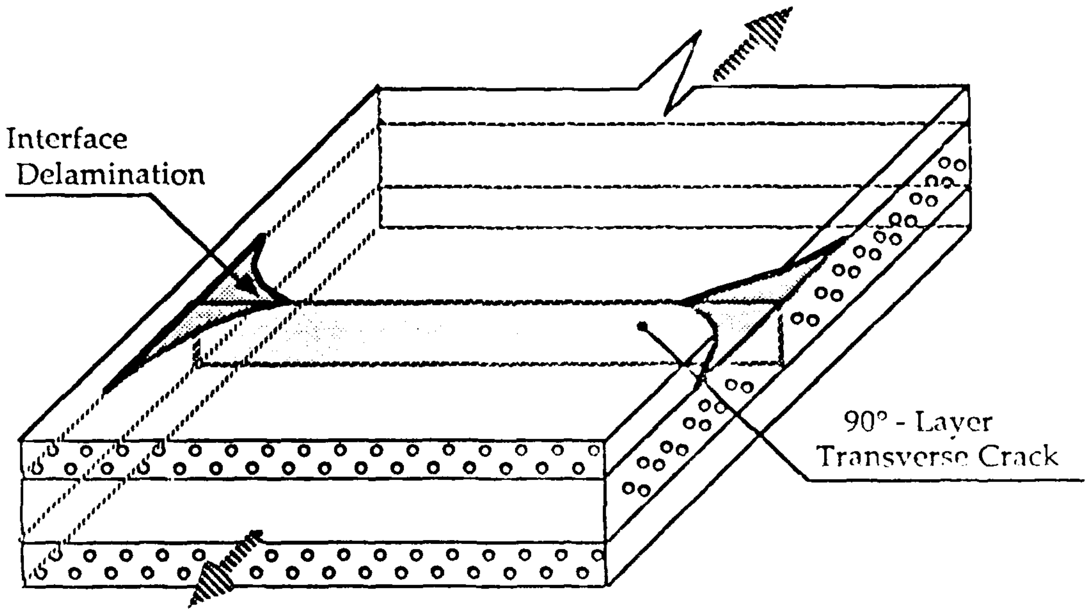
Material properaties and loading information for each layer are furnished in the following manner:

Layer 1 (90° layer)

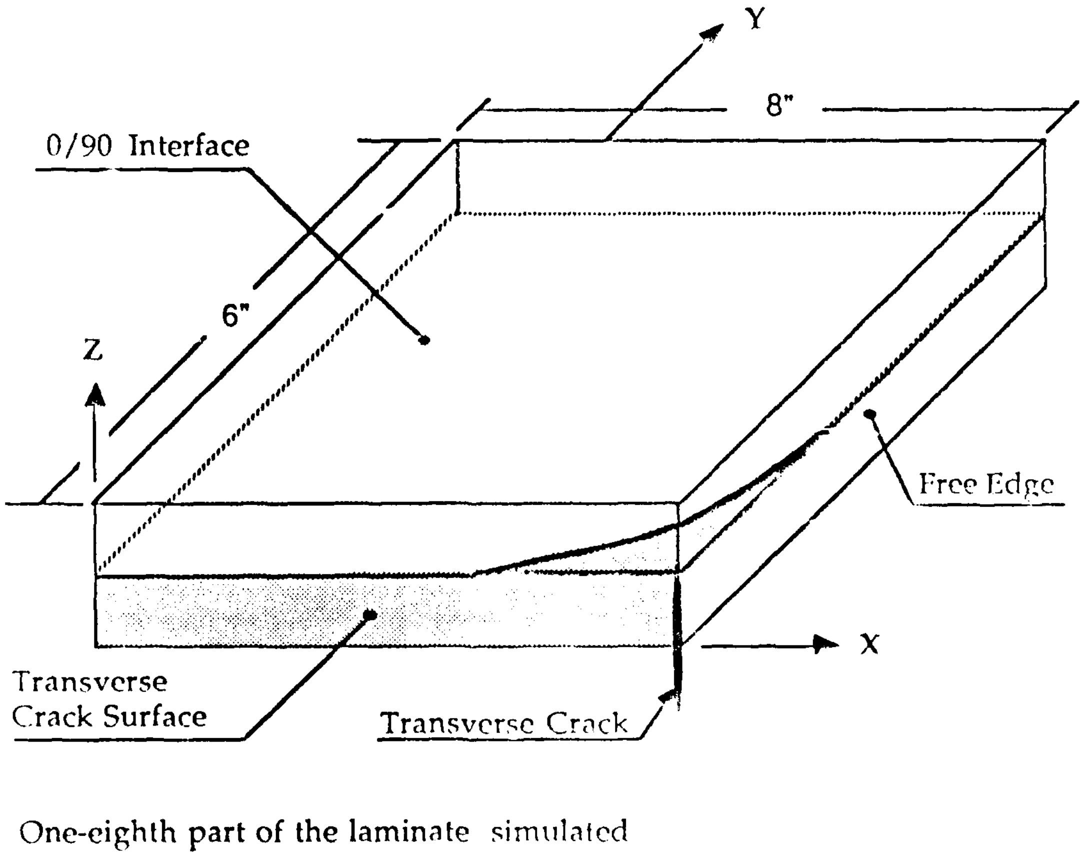
$E_1 = 21.0 \times 10^6$ psi

$E_t = 1.7 \times 10^6$ psi

$E_z = 1.7 \times 10^6$ psi



(a) Transverse Crack/Free Edge Induced Delamination



(b) One-eighth part of the laminate simulated

FIG. 6.1 THE ISOMETRIC VIEWS OF THE EXAMPLE PROBLEM

$\nu_{lt} = 0.30$

$\nu_{lz} = 0.30$

$\nu_{tz} = 0.54$

$G_{lt} = 0.94\text{x}10^{6}$ psi

$G_{lz} = 0.94\text{x}10^{6}$ psi

$G_{tz} = 0.50\text{x}10^{6}$ psi

$\alpha_{l} = 0.20\text{x}10^{-6}$ /°F

$\alpha_{t} = 16.0\text{x}10^{-4}$ /°F

$\alpha_{z} = 16.0\text{x}10^{-4}$ /°F

Layer 2 (0° layer)

The same properties as above.

A uniform displacement of 0.001" is applied in y-direction simulating a constant strain loading and no thermal loading is applied (temp.=0). The delamination is assumed to take place between layer 1 and layer 2 starting at the outer edge at the intersection of free edge and transverse crack. Boundary conditions are provided to make x=0, y=0 and z=0 symmetric surfaces.

The initial finite element mesh without double nodes is as shown in the Figure 6.2.

6.2 PREPROCESSOR INPUT DATA

Group I

```
8 node El.[02/902]s; delam- MECH load; 3x3x5 MESH-man.inp (10/30/87)
```

In this first group the heading to be printed is given on one card.

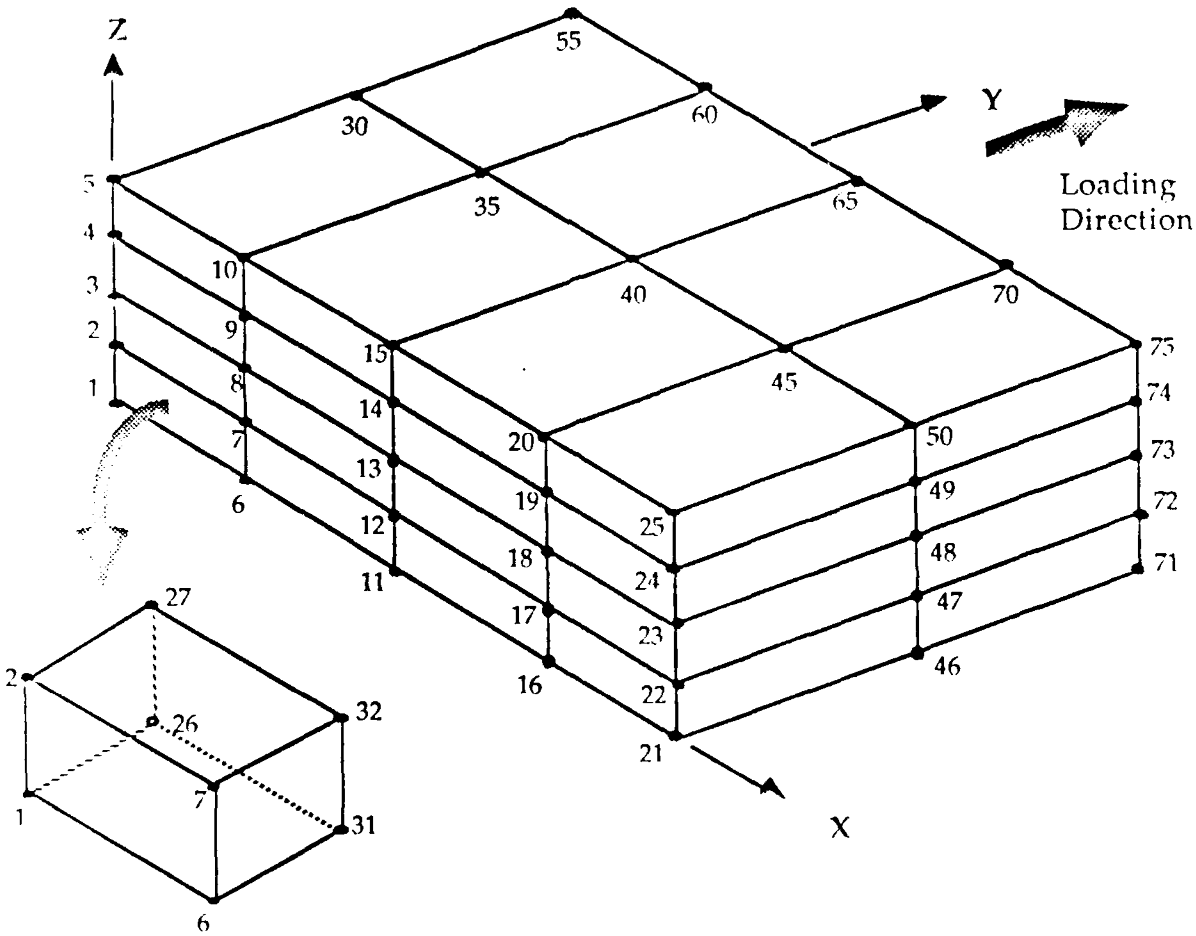


FIG. 6.2 INITIAL FINITE ELEMENT MESH WITHOUT DOUBLE NODES

Group II

```
------------------------------------------------------------------
8
5,3,5,0.0
0.0, 2.0, 4.0, 6.0, 8.0
0.0, 3.0, 6.0
0.0, 0.5, 1.0, 1.5, 2.0
------------------------------------------------------------------
```

The first card indicates that 8-node brick element is to be used. The number of coordinates in x, y and z directions are 5, 3 and 5 respectively. These are given in the second card. The fourth entry in this card is 0.0 and it indicates that there is no hole (hole radius =0.0). The values of x, y and z coordinates are given in the subsequent three cards. No data termination card is required for this set of data.

Group III

```
-----------------------------------------------------------------
 1, 300.0, 75, 1
-1,  0.0,  0, 0
 1  300.0, 32, 1
-1,  0.0,  0, 0
 1, 1, 32, 1
-1, 0,  0, 0
 1, 1, 16, 1
17, 2, 32, 1
-1, 0,  0, 0
 2, 16
18, 32
-1,  0
---  -------------------------------------------------------------
```

In this group, the cards 2, 4, 6, 9 and 12 are for data termination. The first card indicates that all the nodes from 1 to 75 in increments of 1 have a temperature of 300.0 °F. Similarly, the third card is for elements which signifies that stress free temperature for elements from 1 to 32 in increments of 1 is 300.0 °F. The material serial number to which each element belongs to is given in 5th card. In the present problem, two layers of laminate are made up of same material. So, in this card, it is given that elements 1 to 32 (in increments of 1) belong to material set 1.

However, the two layers have different orientations and they are indicated in the 7th and 8th cards. The elements 1 to 16 have the material axis orientation set 1 and 17 to 32 have set 2.

The 10th and 11th cards are to take advantage of the set of identical elements made of the same material. The first card denotes that elements 2 thru 16 are identical to element no. 1 and the same element stiffness matrix is used. Similarly, the 2 nd card denotes that elements 18 thru 32 are the same as the preceding element no. 17.

Group IV

```
0,2
```

This card is for split or notch simulation. The first entry (0) indicates that there are 0 double nodes for split generation. The second entry 2 is for split plane parallel to xz-plane. Since the number of double nodes are zero, the value of second entry can be 1, 2 or 3 and no split will be generated.

Group V

```
15
3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, 63, 68, 73
0.0., 8.0, 0.0, 6.0, 1.0, 2.0
```

The information of nodes which are to be doubled is given in this set of cards. The first card says that there are 15 nodes to be doubled and the succeeding card gives the original numbers of the nodes which are to be doubled. The last card gives limits of the coordinates of the solid in which the second set of double nodes are to be placed.

Group VI

```
------------------------------------------------------------
  EL    1       21.0E6
  ET    1       1.7E6
  EZ    1       1.7E6
NULT    1         0.3
NULZ    1         0.3
NUTZ    1         0.54
 GLT    1       0.94E06
 GLZ    1       0.94E06
 GTZ    1       0.50E06
ALFL    1       0.2E-6
ALFT    1       0.16E-4
ALFZ    1       0.16E-4
-1

1  5 25 55
2 25 75  5
-1
------------------------------------------------------------
```

This set of cards will furnish data regarding the material properties. These properties can be given in any order. The last three cards are to define 2 sets of material principal axes orientations.

Group VII

```
------------------------------------------------------------
-1
------------------------------------------------------------
```

This set of cards is to specify force boundary conditions. In this particular problem a single -1 card signify that there are no force boundary conditions.

Group VIII

```
------------------------------------------------------------
    4   UY   0.0                         24    5
    5   UY   0.0                         25    5
    1   UX   0.0                         51   25
    2   UX   0.0                         52   25
    3   UX   0.0                         53   25
    4   UX   0.0                         54   25
    5   UX   0.0                         55   25
    1   UZ   0.0                         21    5
   26   UZ   0.0                         46    5
   51   UZ   0.0                         71    5
   51   UY   0.001                       75    5
   -1
------------------------------------------------------------
```

The displacement boundary conditions are prescribed in this set of cards. For example, the first card specifies the y-component of displacement as 0.0 for the nodes from 4 thru 24 at 5 node intervals. That is, y-component displacement of nodes 4, 9, 14, 19, 24 have 0.0 value. The boundary conditions of other nodes are prescribed in the succeeding cards of this last set. In this set the original node numbers are to be given and they will be modified using the double nodes information given in Group V

6.3 MODIFICATIONS OF PREPROCESSOR OUTPUT DATA

The output of the preprocessor program will be two data files if NSD is not equal to zero. The file KSAPIN.DAT will consist most of the data necessary to run the main code, KSAP II. The other file FOR010.DAT will contain the information about the modified numbers of the double nodes and their original node numbers on the split plane. These are only for reference and do not appear in the modifications of KSAPIN.DAT.

The original node numbers related to delamination are listed towards the end of KSAPIN.DAT. Both the double nodes of each pair will have the same displacements before the crack passes through them, and both these will have free force boundary conditions once the crack passes through that pair and they are separated. However, some of these pairs lying on y=0 plane behave differently.

The double nodes (3,4),(9,10),(15,16),21,22), and (27,28) are located on the symmetric plane which also happens to be the plane of transverse crack. The y-component of the displacement (Uy) of the node in each pair before they are opened have the same value. When the crack passes through

that node, the two nodes will be separated. The y-component of the force (Fy) of the bottom node will be zero whereas the top node will be still on the symmetric plane (y=0) and hence will have a displacement boundary condition (Uy=0). It may also be observed that the other two components of the forces (Fx,Fz) will become zero for both the nodes of the pair once the crack passes through that pair. To facilitate these two types of degrees of freedom of the double nodes the input data of KSAP II has to be supplemented with the data as shown below.

```
10
 3  2 0.0
 4  2 0.0
 9  2 0.0
10  2 0.0
15  2 0.0
16  2 0.0
21  2 0.0
22  2 0.0
27  2 0.0
28  2 0.0
15
 3  4 1 0 1
 9 10 1 0 1
15 16 1 0 1
21 22 1 0 1
27 28 1 0 1
33 34 1 1 1
39 40 1 1 1
45 46 1 1 1
51 52 1 1 1
57 58 1 1 1
63 64 1 1 1
69 70 1 1 1
75 76 1 1 1
81 82 1 1 1
87 88 1 1 1
```

The number (10) in the first data card denotes the total number of y-degrees of freedom of double nodes (5x2=10) on the symmetric plane (y=0). The following 10 data cards input the details of the node number, degree of freedom (1 for Ux, 2 for Uy and 3 for Uz) and the value of the displacement. All these data corresponds to that value before the nodes

are opened. Following this will be the comprehensive data input for the overal double nodes. The number 15 denotes that there are 15 pairs of double nodes whose details are given in the 15 succeeding cards. The first two numbers in each card (for example, 3 and 4) are two node numbers in each pair. The three following numbers (1's or 0's) will describe the behavior of the double nodes in x,y,z degrees of freedom respectively when the nodes are opened. The number 1 for any degree of freedom signifies that the nodes will have the same displacement before opening and will have zero nodal force in that degree of freedom after opening. Number 0 for any particular degree of freedom signifies that the nodes will not behave in the above manner with regard to that degree of freedom. Thus the degrees of freedom (y) for the double nodes on the symmetric plane, y=0 which are described in the previous set will have zeros in this set of data. For example, the nodes 3 and 4 will have Uy=0 corresponding to y-degree of freedom as they are specified to behave in another manner in the preceding set of data. The nodes 27 and 28 will have all 1's as they are located away from the plane y=0.

After furnishing the above data regarding the degrees of freedom of the double nodes, comes the data regarding the opening of the double nodes thus simulating a crack propagation as shown below:

```
------------------------------------------------------------
 0  0  0
    1   32          ! Free format
27  0  2
27 28  1
27 28  3
 0  0  0
    1   32
21  0  2
21 22  1
21 22  3
57 58  1
57 58  2
57 58  3
```

```
0 0 0
  1  32
9999 9999 0
```

At any step (iteration) the crack can be made to pass through one or more number of double nodes. Each data card consists of three numbers. The last number corresponds to the degree of freedom which is relaxed, that is, which will have free force boundary condition. If there is a zero as the second number then the first number should be a node number of the double nodes on the symmetric plane y=0 whose data is specified in the first set. For example, the '27 0 2' specifies that the y-degrree of freedom (2) of node no. 27 (which is a node in 1st set of data) is relaxed (Fy=0). That is, this node is free to move in y-direction. If the second number is also non zero then the first two numbers correspond to the two nodes of a pair of double nodes and as explained above the third number specifies the degree of freedom in which these two nodes are free to move. Thus the data specifies that the crack passes through the pair of nodes 27 and 28 and these two nodes are free to move in x, z directions whereas in the y-direction only node no. 27 is free to move. That implies that node 28 will have the earlier specified diplacement (Uy=0). All three zeros signify the end of the crack opening data for that step. Thus when it is desired to calculate the stresses and displacements before any crack is simulated it is necesssary to place this card (0,0,0) as in line 1. Immediately following this card in each step a selective stress print option can be given. The two numbers signify the range of elements for which the stress print out is desired. If the first two numbers in the crack opening data are prescribed as 9999 and 9999 then that signals the termination of crack propagation sequence.

6.4 OUTPUT OF KSAP II PROGRAM

As can be seen the output of KSAP II is self explanatory. To start with, it consists of all the mesh details regarding the nodes, coordinates, the degrees of freedom, elements etc.. It also furnishes information of the material properties used and the material number to which each element belongs. The output also provides some details about total number of equations, bandwidth, number of blocks etc..

Then the results will be output as the crack is simulated. For each step the nodal displacements and element stresses (at the center of the element) are printed as desired in the input data. Usually, these results are output starting from no crack state (STEP 0) and crack can be simulated opening one or more nodes at each step. At each succeeding step (STEP 1,2,....) the energy released is calculated and is printed immediately after nodal displacements and forces. If the energy release rate is desired then it can be calculated by dividing these energies released by the incremental crack areas.

APPENDIX - A

LISTING OF THE PREPROCESSOR

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*                                                            *
*            THIS CODE  GENERATES INPUT DATA FOR 'KSAP II'   *
*                                                            *
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!
!                  version    :  September 1987
!
!            Reads from Input file  and writes to KSAPIN.DAT
!
!
! NOTE:       1)  Modified to generate 21-node element directly from
!                 the regular mesh WITHOUT ELEMENT GENERATION CODES
!
!                 Assign  21 / 8 (= NTYPE ) as the 2nd card in the
!                 input data file
!
! CAUTION:
!             2)  The nodal input order for the elements (NTYPE=21)
!                 SHOULD be as in SAPIV MANUAL
!             3)  Double nodes region is to be given
!                 (xl,xu, yl,yu, and zl,zu in that order)
!
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!
!      R(),NR()--- temporary storage of data read
!      NOS(I)  --- number of sets to be generated at i-th  level
!      NOE(I)  --- number of elements in the set to be generated
!      NINC(I) --- increasment of node number
!      NN(I,J) --- node numbers of element -i
!      IDM     --- i.d. number of matl.
!      MAT(I)  --- 1 : stiffness matrix of element -i is the same as previous
!                  0 : not the same as last
!      MAXES(I)--- I.D.no. of matl. axis orientation set(see SAP4 manual)
!      TZ(I)   --- Stress free reference temperature for element -I
!      X(I)  } --- coordinets of node -I
!      Y(I)  }
!      Z(I)  }
!      XX(I) } --- coor. increments in level generation
!      YY(I) }
!      ZZ(I) }
!      T(I)    --- Nodal temperature
!      E11 ..  --- Material properties
!      IX(I,J) --- Degree of freedom code , node -i, freedom -j
!      NBD()   --- No. of boundary element FOR dISPLACEMENT
!      ND(I,4) --- 4 nodes used for defining the direction of displ.
!      KD()    --- Means displ. if equals to 1
!      KR()    --- ,,   rotation if equals to 1
!      NBF( )  --- node no. of force boundary
!      FX()  } --- concentrated force
!      FY()  }
!      FZ()  }
!      L(4)    --- Node difining vector criss multiplied to give
```

```
!                    the direction of displ. or rotation
!       VD()    --- Value of displ.
!       VR()    --- Value of rotation
!       LAX(2,3) -- 3 nodes (number) to define matl. princ. axis
!                    (max. of 2 sets)
!       HED(18)  -- Reading information as first line of output
!       MORT()   -- material axes orientation set number
!       NI()  } --- definition of material principle axes
!       NJ()  }
!       NK()  }
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        CHARACTER*30 FILNAM
        CHARACTER*1 MESH
        DIMENSION R(7),NR(15),NRR(21),NOS(3),NOE(3),NINC(3),
     .    XX(20),YY(20),ZZ(15),XMESH(4000),YMESH(4000),

     .    X(4000),Y(4000),Z(4000),IX(4000,6),T(4000),mip(4000,2),

     .    IDM(3000),MAT(3000),NN(3000,21),MAXES(3000),TZ(3000),

     .    E11(4),E22(4),E33(4),ANU12(4),ANU13(4),ANU23(4),
     .    G12(4),G13(4),G23(4),NBD(200),ND(200,4),KD(200),
     .    L(4),VD(200),VR(200),KR(200),HED(18),
     .    NOND(200),XD(200),YD(200),ZD(200),  nons(400),
     .    ALP1(4),ALP2(4),
     .    ALP3(4),MORT(10),NI(10),NJ(10),NK(10),
     .    FX(200),FY(200),FZ(200),NBF(200),ID1(200)

c--      mip ( ) - the corresponding new nodal numbers
C

        PI=3.1415926535897932
        eps=1.0e-04
        WRITE (5,*) '...ENTER INPUT FILE NAME.....'
        READ (5,55) FILNAM
        IRD=56          !READ TAPE NO.
        OPEN (UNIT=IRD,FILE=FILNAM,STATUS='OLD')
        READ(IRD,25) (HED(I),I=1,18)
     25 FORMAT(18A4)

!Read whether the element is 8- or 21 noded
c        NTYPE=21
c        NTYPE=8
        read (IRD,*) ntype
        NELTYP=1
        INTRS=2          !Integration order for r,s - coordinates
        INTT=2           !Integration order for t- direction
        IF (NTYPE.EQ.21) THEN
        INTRS=4
        INTT=2
        END IF
C
C!!!!! FOLLOWING STATEMENTS ARE FOR VARYING GRID GENERATION.
```

```
C
C      NODE GENERATION
       READ(IRD,*),NONX,NONY,NONZ,RHOLE
       READ (IRD,*)(XX(I),I=1,NONX)
       READ (IRD,*)(YY(I),I=1,NONY)
       READ (IRD,*)(ZZ(I),I=1,NONZ)

c--    correction for inside nodal coordinates for 21-node el.

       if (ntype.eq.21) then
       do 1444 i=1,nonx
 1444  if (mod(i,2).eq.0) xx(i)=(xx(i-1)+xx(i+1))/2
C       type *,(xx(i),i=1,nonx)
       do 1445 i=1,nony
 1445  if (mod(i,2).eq.0) yy(i)=(yy(i-1)+yy(i+1))/2
C       type *,(yy(i),i=1,nony)
       do 1446 i=1,nonz
 1446  if (mod(i,2).eq.0) zz(i)=(zz(i-1)+zz(i+1))/2
C       type *,(zz(i),i=1,nonz)
       end if

       IF (RHOLE.GT.eps)CALL HOLE (RHOLE,NONX,NONY,NONZ,
      .XX,YY,zz,X,Y,Z)
       IF (RHOLE.GT.eps) GO TO 444

       DO J=1,NONY
       DO I=1,NONX
       DO K=1,NONZ
       N=(J-1)*NONX*NONZ+(I-1)*NONZ+K
       X(N)=XX(I)
       Y(N)=YY(J)
       Z(N)=ZZ(K)
       END DO
       END DO
       END DO

C-     END OF POLAR MESH GENERATION

 444     NTON=NONX*NONY*NONZ

C--      MESH PLOTTING OPTION ON HP PLOTTER-------
         WRITE (5,*) 'Do you need MESH plot Original nodes?..(Y/N)'
         READ (5,55) MESH
 55      FORMAT (A)
         IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL
       . MESHPL (X,Y,NONX,NONY,NONZ,MIP,1,ntype)
         DO I=1,NTON
         XMESH(I)=X(I)
         YMESH(I)=Y(I)
         END DO
         IF (NN(NTOE,7).GT.NTON) THEN
         DO 666 I=1,NTOE
         DO 666 J=1,NTYPE
 666     IF (NN(I,J).GT.NTON) NN(I,J)=NN(I,J)-NTON
```

```
          END IF

148    READ(IRD,*) N,TEMP,NEND,INC
       IF(N.EQ.-1) GO TO 149
       DO I=N,NEND,INC
       T(I)=TEMP
       END DO
       GO TO 148

 149   continue
300    CONTINUE
C---------------------------------------------------------------------------------
C      ELEMENT GENERATION
C
       N1=NONX
       N2=NONY
       N3=NONZ
       N13=N1*N3

       IF(NTYPE.EQ.8) THEN

       NOS(3) =N3-1               ! third level gener.code
       NINC(3)=1
       NOE(3) =(N2-1)*(N1-1)
       NOS(2) =N2-1               !  second level gener. code
       NINC(2)=N1*N3
       NOE(2) =N1-1
       NOS(1) =N1-1               !  first level gener. code
       NINC(1)=N3
       NOE(1) =1

       NRR(1)=2+  N3+  N13
       NRR(2)=2+       N13
       NRR(3)=2
       NRR(4)=2+  N3
       NRR(5)=1+  N3+  N13
       NRR(6)=1     +  N13
       NRR(7) =1
       NRR(8)=1+  N3

       ELSE IF(NTYPE.EQ.21) THEN

       NOS(3) =(N3-1)/2               ! third level gener.code
       NINC(3)=2
       NOE(3) =(N2-1)*(N1-1)/4
       NOS(2) =(N2-1)/2               !  second level gener. code
       NINC(2)=2*N1*N3
       NOE(2) =(N1-1)/2
       NOS(1) =(N1-1)/2               !  first level gener. code
       NINC(1)=2*N3
       NOE(1) =1

       NRR(1) =3+2*N3+2*N13
       NRR(2) =3     +2*N13
```

```
      NRR(3) =3
      NRR(4) =3+2*N3
      NRR(5) =1+2*N3+2*N13
      NRR(6) =1      +2*N13
      NRR(7) =1
      NRR(8) =1+2*N3

      NRR(9 )=3+  N3+2*N13
      NRR(10)=3     +  N13
      NRR(11)=3+  N3
      NRR(12)=3+2*N3+  N13
      NRR(13)=1+  N3+2*N13
      NRR(14)=1     +  N13
      NRR(15)=1+  N3
      NRR(16)=1+2*N3+  N13
      NRR(17)=2+2*N3+2*N13
      NRR(18)=2+2*N13
      NRR(19)=2
      NRR(20)=2+2*N3

      NRR(21)=2+  N3+  N13
      END IF
      NO=0
      DO 132 I=1,NTYPE
 132  NN(NO+1,I)=NRR(I)

!     Perform element generation
      DO 138  M=1,3
      DO 138  K=2,NOS(M)
      DO 138  I=1,NOE(M)
      N=NO+I+NOE(M)*(K-1)
      DO 1375 J=1,NTYPE
 1375 NN(N,J)=NN(NO+I,J)+NINC(M)*(K-1)
 138  continue
      NO=N
      NTOE=NO
!       DO 1415 IE=1,NTOE
!      TYPE *
!      TYPE *,' **** ELEMENT ' ,IE
!1415      TYPE 105, (NN(IE,J),J=1,NTYPE)
 105  FORMAT(8I5)


!     Read element stress free temperature
 158  READ(IRD,*) N,TEMP,NEND,INC
      IF(N.EQ.-1) GO TO 159
      DO 1585 IE=N,NEND,INC
 1585 TZ(IE)=TEMP
      GO TO 158
 159  continue

!     Read element material identification number
 1458 READ(IRD,*) N,MATRL,NEND,INC
      IF(N.EQ.-1) GO TO 1459
```

```
      DO 14585 IE=N,NEND,INC
14585 IDM(IE)=MATRL
      GO TO 1458
1459  continue
C ---MAXES(I)______MATL. AXIS ORIENT.:
!     Read element material axis orientation identification number
1658  READ(IRD,*) N,MORTT,NEND,INC
      IF(N.EQ.-1) GO TO 1659
      DO 16585 IE=N,NEND,INC
16585 MAXES(IE)=MORTT
      GO TO 1658
1659  continue
!     Read if the stiffness matrix of an element (or a group of elements)
!     is same as the previous element
      DO 145 I=1,NTOE
145   MAT(I)=0
176   READ(IRD,*) M1,M2
c     WRITE(111,*) 'M1,m2',m1,m2
      IF(M1.EQ.-1)GO TO 1765
      DO 1762 I=m1,M2
1762  MAT(I)=1
      GO TO 176
1765  continue
C-----------------------------------------------------------------------
C     BOUNDARY CONDITION GENERATION
c     Following  statement are to fix all rotations when
c     only translational d.o.f.  (eltype #8 is used)
      DO 2010 I=1,NTON
      DO 2010 J=1,6
      IF(J.LE.3) IX(I,J)=0
      IF(J.GE.4) IX(I,J)=1
2010  continue
!     While generating 21-node elements some nodes are at the center of faces
!     These degrees of freedom will be removed below:
!     Nodes not used in defining any element are found
!     and all its d.o.f. set equal to 1 (i.e., eliminated)
      DO IE=1,NTOE
      DO IN=1,NTYPE
      II=NN(IE,IN)
      IX(II,6)=10          !to identify which nodes are used
      end do
      end do
      DO IND=1,NTON
      IF(IX(IND,6).EQ.10)THEN
      IX(IND,6)=1
      ELSE
      DO IDG=1,6
      IX(IND,IDG)=1
```

```
      END DO
      END IF
      end do
!!!!!! FOR DOUBLING NODES !!!!!!!!!!!!!!!!!!!!!!!!!!!!
      do ip=1,nton
      mip(ip,2)=0
      end do
!     For Splitting..............
      NSD=0
      READ(IRD,*) NSD,IDIR    ! No. of nodes to be doubled, Direction Vector
      IF(NSD.NE.0) THEN
      NSD1=0
686   READ (IRD,*) N,NEND,INC
      IF (N.EQ.-1) GO TO 688
      DO 687 I=N,NEND,INC
      NSD1=NSD1+1
687   NONS(NSD1)=I
      GO TO 686
!      READ(IRD,*) (NONS(I),I=1,NSD)    !original node # to be doubled
688   IF (NSD1.NE.NSD) NSD=NSD1         !check on total no. of d.n's
c--   to delete face center no. from double node list.......
      IF (NTYPE.EQ.21) CALL DELETE(NONX,NONY,NONZ,NSD,NONS)
      do i=1,nsd
      ia=nons(i)
      mip(ia,2)=1
      end do
      end if
      NTD=0
      READ(IRD,*) NTD    ! Number of nodes to be doubled

!     Define the zone of delamination
c     store coord. of the nodes to be doubled
      IF(NTD.NE.0) THEN
      READ(IRD,*) (NOND(I),I=1,NTD)   !original node # to be doubled
c--   to delete face center no. from double node list.......
      IF (NTYPE.EQ.21) call delete(nonx,nony,nonz,ntd,nond)
      DO I=1,NTD
      IA=NOND(I)
      MIP(IA,2)=1
      END DO
      READ(IRD,*) XL,XU,YL,YU,ZL,ZU
      DO I=NTD,1,-1
      XD(I)=X(NOND(I))
      YD(I)=Y(NOND(I))
      ZD(I)=Z(NOND(I))
      END DO
      END IF
C      if (ntd+nsd.ne.0) then
      IP=0
      DO I=1,NTON
      IP=IP+1
      MIP(I,1)=IP
      IF(MIP(I,2).EQ.1) THEN
      IP=IP+1
```

```
      MIP(I,2)=IP
      END IF
      END DO
C      end if
      IF (NSD.NE.0)
     .call split (NTON,nsd,idir,nons,mip,x,y,z,T,ntype,nos,NN,ntoe)
!     For Delamination............
C      DO I=1,NTON
C      TYPE *,I,(MIP(I,J),J=1,2)
C      END DO

      IF(NTD.EQ.0) GO TO 899
C-   CORRECTIONS FOR DOUBLE NODES IN planes normal to x, y or z -directions
C    FIND KND (THE # OF THE NODE CURRENTLY TO BE DOUBLED)

      DO 525 I=1,NTD
      DO J=NOND(I),NTON

      DIST=SQRT((X(J)-XD(I))**2+ (Y(J)-YD(I))**2+ (Z(J)-ZD(I))**2)
      IF(DIST.LT.0.00001) THEN
      KND=J

C     BEGIN CHANGING NODE NUMBERS & COORD.

c     Change element node numbers if it is > knd by adding 1 to it
      DO 346 JE=1,NTOE
      DO 345 K=1,NTYPE
345   IF (NN(JE,K).GT.KND)  NN(JE,K)=NN(JE,K)+1
346   CONTINUE
C     Also move coords. downstream by one slot and also assign (knd+1)th
!      same as (knd)th
      DO M=NTON,KND,-1
      X(M+1)=X(M)
      Y(M+1)=Y(M)
      Z(M+1)=Z(M)
      T(M+1)=T(M)
      END DO

C--  Allocate nodes of the pair to the appropriate elements depending
!    on which side of the double nodes' plane they (element) lie
      nface=NTYPE
      DO 3455  K=1,NTOE

      IF(NTYPE.EQ.8) THEN      ! Find the coordinates of the center of the
      XC=0.                    ! element to determine if it belongs
      YC=0.                    ! to the delamination zone
      zc=0.
      DO IA=1,8
      III=NN(K,IA)
      XC=XC+X(III)
      YC=YC+Y(III)
      ZC=ZC+Z(III)
      END DO
      XC=XC/8.
```

```
      YC=YC/8.
      ZC=ZC/8.
      ELSE
      III=NN(K,21)
      XC=X(III)
      YC=Y(III)
      ZC=Z(III)
      END IF
!      IF(K.EQ.32) THEN
!      TYPE *, ' ... ELEMENT # ... ', K
!      TYPE *, ' - CENTER:', XC,YC,ZC
!      TYPE *, ' .. BEFORE ...'
!      TYPE *, (NN(K,M),M=1,NFACE)
!      END IF
      DO 3453 M=1,nface
       IF (NN(K,M).EQ.KND
     .     .and.(XC.GT.XL.AND.XC.LT.XU)
     .     .and.(YC.GT.YL.AND.YC.LT.YU)
     .     .and.(ZC.GT.ZL.AND.ZC.LT.ZU))

     . NN(K,M)=KND+1
 3453     END DO
 3455 END DO


      NTON=NTON+1

      go to 525   ! so that the modifcations are done only once
!                   for each double node
      END IF       !  (DIST.LT.0.00001)
      END DO           !J - LOOP
 525  END DO           !I - loop

  899 CONTINUE        !    SKIP IF  NTD=0

      write(5,*)'...Do You Need Mesh with Double Nodes (Y/N)..'
      READ (5,55) MESH
      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL
     .  MESHPL (XMESH,YMESH,NONX,NONY,NONZ,MIP,2,ntype)

      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
     .write(5,*)'...Do You Want to Continue (Y/N)..'
      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
     .READ (5,55) MESH
      IF (MESH.EQ.'Y'.and.MESH.EQ.'y') stop

!!!!!!!!! ABOVE ARE FOR DOUBLING NODES !!!!!!!!!!!!!!!!

!      Material properties
      OPEN (UNIT=9,FILE='KSAPIN.DAT',STATUS='NEW')    !!!!!

      nummat=0
 500  read(IRD,550) A,I,V
      if(i.gt.nummat) nummat=i
```

```
550    FORMAT(A4,I4,4X,G17.7)
       IF(A.EQ.'  EL') E11(I)=V
       IF(A.EQ.'  ET') E22(I)=V
       IF(A.EQ.'  EZ') E33(I)=V
       IF(A.EQ.'NULT') ANU12(I)=V*E22(I)/E11(I)
       IF(A.EQ.'NUTZ') ANU23(I)=V*E33(I)/E22(I)
       IF(A.EQ.'NULZ') ANU13(I)=V*E33(I)/E11(I)
       IF(A.EQ.' GLT') G12(I)=V
       IF(A.EQ.' GLZ') G13(I)=V
       IF(A.EQ.' GTZ') G23(I)=V
       IF(A.EQ.'ALFL') ALP1(I)=V
       IF(A.EQ.'ALFT') ALP2(I)=V
       IF(A.EQ.'ALFZ') ALP3(I)=V
       IF(A.EQ.'-1') GO TO 560
       GO TO 500
 560   DO I=1,10
       READ(IRD,570) (NR(J),J=1,4)
 570   FORMAT(4I5)
       IF(NR(1).EQ.-1) GO TO 579
       MORT(I)=NR(1)
       NI(I)=NR(2)
       NJ(I)=NR(3)
       NK(I)=NR(4)
       END DO
 579   NORTHO=I-1
!THE BELOW PORTION IS MODIFICATION TO FIND DIRECTION VECTOR
!above one does not work for hole problems (polar mesh).--
C       TYPE *, 'NTON',NTON
       DO 991 I=1,NTON
       DO 992 J=1,NTON
       IF (J.EQ.I) GO TO 992
       IF (ABS(Y(J))-ABS(Y(I)).GT.1.E-06) GO TO 992
       IF (ABS(Y(J)-Y(I)).GT.1.E-06) GO TO 992
       IF (ABS(Z(J))-ABS(Z(I)).GT.1.E-06) GO TO 992
       IF (ABS(Z(J)-Z(I)).GT.1.E-06) GO TO 992
       IF (X(J)-X(I).LT.0.0) GO TO 992
       DO 993 K=1,NTON
       IF (K.EQ.I.OR.K.EQ.J) GO TO 993
       IF ((ABS(X(K))-ABS(X(I))).GT.1.E-06) GO TO 993
       IF (ABS(X(K)-X(I)).GT.1.E-06) GO TO 993
       IF ((ABS(Z(K))-ABS(Z(I))).GT.1.E-06) GO TO 993
       IF (ABS(Z(K)-Z(I)).GT.1.E-06) GO TO 993
       IF ((Y(K)-Y(I)).LT.0.0) GO TO 993
       DO 994 LLL=1,NTON
       IF (LLL.EQ.I.OR.LLL.EQ.J.OR.LLL.EQ.K) GO TO 994
       IF ((ABS(X(LLL))-ABS(X(I))).GT.1.E-06) GO TO 994
       IF (ABS(X(LLL)-X(I)).GT.1.E-06) GO TO 994
       IF ((ABS(Y(LLL))-ABS(Y(I))).GT.1.E-06) GO TO 994
       IF (ABS(Y(LLL)-Y(I)).GT.1.E-06) GO TO 994
       IF ((Z(LLL)-Z(I)).LT.0.0) GO TO 994
```

```
        LO=I    !y & z coord. same as J, xcoord. diff.
        LX=J    !y & z coord. same as I, xcoord. diff.
        LY=K    !.ne.I or J, x & z coord. same as I and y is diff.
        LZ=LLL  !.ne. I,J or K, x & y coord. same as I and z diff.
        GO TO 995
994     CONTINUE
993     CONTINUE
992     CONTINUE
991     CONTINUE
995     CONTINUE

C-- READ & GENERATE CONCENTRATED LOAD DATA:

        NTBF=0
        NINCC=1
640     READ(IRD,710) N1,A,V,N2,NINCC
        IF(N1.EQ.-1) GO TO 699

        DO I=N1,N2,NINCC
        NTBF=NTBF+1
        NBF(NTBF)=I
        if(A.EQ.'  FX') FX(NTBF)=V
        if(A.EQ.'  FY') FY(NTBF)=V
        if(A.EQ.'  FZ') FZ(NTBF)=V
        END DO
        GO TO 640
    699 CONTINUE
C
C       READ & GENERATE DISPL. B.C.E. DATA
C
        NTBD=0
        NINCC=1
    700 READ(IRD,710) N1,A,V,N2,NINCC
    710 FORMAT(I6,1X,A4,1X,F10.0,12X,2I6)
        IF(N1.EQ.-1) GO TO 799


        IF(V.EQ.0) THEN
        DO I=N1,N2,NINCC
        IF(A.EQ.'  UX') IX(I,1)=1
        IF(A.EQ.'  UY') IX(I,2)=1
        IF(A.EQ.'  UZ') IX(I,3)=1
        END DO

        ELSE
        NELTYP=2
        V1=V
        V2=0
        JD=1
        JR=0
        IF(A.EQ.'  UX') then
        L(1)=LO
        L(2)=LY  !LO -> LY ALONG Y DIR.
        L(3)=LO  !LO -> LZ ALONG Z DIR.
```

```
      L(4)=LZ
      else if(A.EQ.'  UZ') then
      L(1)=LO
      L(2)=LX
      L(3)=LO
      L(4)=LY
      else if(A.EQ.'  UY') then
      L(1)=LO
      L(2)=LZ
      L(3)=LO
      L(4)=LX
      END IF

      DO I=N1,N2,NINCC
      NTBD=NTBD+1
      NBD(NTBD)=I
      DO J=1,4
      ND(NTBD,J)=L(J)
      END DO
      VD(NTBD)=V1
      VR(NTBD)=V2
      KD(NTBD)=JD
      KR(NTBD)=JR
      END DO
      end if
      GO TO 700
  799 CONTINUE

C  writing double nodes to for010.dat, and renumbering displacement
c  boundary elements and force boundary node number

       do i=nton,1,-1
        do ic=1,2
          ip=mip(i,ic)
          if(ip.ne.0) then
            do id=1,6
            ix(ip,id)=ix(i,id)
            end do
          end if
        end do
       end do

      if(ntd+nsd.ge.1) then
      DO K=1,NTBD

      NBDK1=MIP(NBD(K),1)
      NBDK2=MIP(NBD(K),2)
      NBD(K)=NBDK1
      KR(K)=MIP(KR(K),1)
      KD(K)=MIP(KD(K),1)

      IF(NBDK2.NE.0) THEN
      ntbd=ntbd+1
      NBD(NTBD)=NBDK2
```

```
      KR(NTBD)=KR(K)
      KD(NTBD)=KD(K)
      VD(NTBD)=VD(K)
      VR(NTBD)=VR(K)
      DO M=1,4
      ND(NTBD,M)=ND(K,M)
      END DO
      END IF
      END DO

C   RENUMBER NI,NJ,NK (ORIENT. DEFINITION)
      DO K=1,NORTHO
      NI(K)=MIP(NI(K),1)
      NJ(K)=MIP(NJ(K),1)
      NK(K)=MIP(NK(K),1)
      END DO

      end if    !  ntd.ge.1


C      Renumbering the force boundary conditions and adding
c      new force boundary conditions of double nodes if necessary

      DO K=1,NTBF
      if(mip(nbf(k),2).ne.0) THEN
      WRITE (5,*) ' ....D.N at force b.c; NODE no..', NBF(K)
      STOP   'STOPPING due to double node at force boundary cond.'
      END IF
      NBF(K)=mip(NBF(K),1)
      END DO
C
C     OUTPUT NODE DATA
C
      WRITE(9,1100) (HED(I),I=1,18)
 1100 FORMAT(18A4)
      IADOF=NTD*6
!         IADOF===ADDITIONAL D.O.F. DUE TO DOUBLE NODES & DISP.BCS.
      WRITE(9,1101) NTON,NELTYP,IADOF
 1101 FORMAT(I5,I5,4X,'1',14X,'0',15X,I5)


      WRITE(9,11015) (IX(1,J),J=1,6),X(1),Y(1),Z(1),T(1)
11015 FORMAT(4X,'1C',I4,5I5,3F10.4,5X,F10.0)
      KN=0
      WRITE(9,1102) 2,(IX(2,J),J=1,6),X(2),Y(2),Z(2),KN,T(2)
      KNM=0
      DO I=3,NTON-1
      IXM1=ix(i,1)-ix(i-1,1)
      IXM2=ix(i,2)-ix(i-1,2)
      IXM3=ix(i,3)-ix(i-1,3)
      IXM4=ix(i,4)-ix(i-1,4)
      IXM5=ix(i,5)-ix(i-1,5)
      IXM6=ix(i,6)-ix(i-1,6)
      DXM=X(I)-X(I-1)
```

```
      DYM=Y(I)-Y(I-1)
      DZM=Z(I)-Z(I-1)
      DTM=T(I)-T(I-1)
      IXP1=ix(i+1,1)-ix(i,1)
      IXP2=ix(i+1,2)-ix(i,2)
      IXP3=ix(i+1,3)-ix(i,3)
      IXP4=ix(i+1,4)-ix(i,4)
      IXP5=ix(i+1,5)-ix(i,5)
      IXP6=ix(i+1,6)-ix(i,6)
      DXP=X(I+1)-X(I)
      DYP=Y(I+1)-Y(I)
      DZP=Z(I+1)-Z(I)
      DTP=T(I+1)-T(I)
      KN=0
      IF(IXM1.EQ.IXP1.AND.IXM2.EQ.IXP2.AND.IXM3.EQ.IXP3.AND.
     .   IXM4.EQ.IXP4.AND.IXM5.EQ.IXP5.AND.IXM6.EQ.IXP6.AND.
     .   DXM.EQ.DXP.AND.DYM.EQ.DYP.AND.DZM.EQ.DZP.AND.
     .   DTM.EQ.DTP) KN=1
      IF(KN.EQ.0)
     .WRITE(9,1102) I,(IX(I,J),J=1,6),X(I),Y(I),Z(I),KNM,T(I)
      KNM=KN
 1102 FORMAT(I5,6I5,3F10.4,I5,F10.0)
      END DO
      i=nton
      WRITE(9,1102) I,(IX(I,J),J=1,6),X(I),Y(I),Z(I),KN,T(I)
C
      if (neltyp.gt.1) then    !     OUTPUT FOR B.C.E. #7
C
      WRITE(9,1201) NTBD
 1201 FORMAT(4X,'7',I5/8X,'1.')
      DO I=1,NTBD
      WRITE(9,1202) NBD(I),(ND(I,J),J=1,4),KD(I),KR(I),VD(I),VR(I)
 1202 FORMAT(7I5,5X,2F10.7,' 0.100E+21  ')
      END DO
      end if
C
C     OUTPUT ELEMENT DATA
C
      MAXNOD=NTYPE
      NOPSET=1     !Number of sets of data requesting stress output
      WRITE(9,1300) NTOE,NUMMAT,NORTHO,MAXNOD,NOPSET,INTRS,INTT
 1300 FORMAT(4X,'8',I5,i5,4X,'0',I5,5X,I5,3I5)

c     I=1
      do i=1,nummat
      WRITE(9,1301) I,TZ(1),E11(I),E22(I),E33(I),ANU12(I),ANU13(I),
     .ANU23(I),G12(I),G13(I),G23(I),ALP1(I),ALP2(I),ALP3(I)
 1301 FORMAT(I5,4X,'1',20X,'AXIS#1==0-LAYER; AXIS#2==90-LAYER.'/
     *f10.0,3f10.0,3f10.4/3f10.0,3f10.7)
ccc     WRITE(9,13010) T(1),E11(I),E22(I),E33(I),ANU12(I),ANU13(I),
ccc    .ANU23(I),G12(I),G13(I),G23(I),ALP1(I),ALP2(I),ALP3(I)
13010 FORMAT(F10.0,3F10.0,3F10.4/3F10.0,3F10.7)
      END DO

```

```
      DO I=1,NORTHO
      WRITE(9,13011) MORT(I),NI(I),NJ(I),NK(I)
13011 FORMAT(4I5)
      END DO

13005 FORMAT(4I5/4I5/3I5)

      READ(IRD,*) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
      WRITE(9,13008) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
13008 FORMAT(7I5)
      TA=1.0
      IF(T(1).EQ.TZ(1)) TA=0.0
      WRITE (9,13012)  TA
13012 FORMAT(///E10.0/)
      IOP=1       ! I.D.# OF STRESS OUTPUT LOCATION SET
      ISKIPO=0

      DO I=1,NTOE
      KGM=0
      if(i.gt.1.and.
     .IDM(I).EQ.IDM(i-1).AND.MAXES(I).EQ.MAXES(i-1).AND.IOP.EQ.IOP
     .    .AND.TZ(I).EQ.TZ(i-1).AND.MAT(I).EQ.MAT(i-1))  then
      KGM1=NN(I,7)-NN(I-1,7)
      KGM2=NN(I,8)-NN(I-1,8)
      KGM3=NN(I,5)-NN(I-1,5)
      KGM4=NN(I,6)-NN(I-1,6)
      KGM5=NN(I,3)-NN(I-1,3)
      KGM6=NN(I,4)-NN(I-1,4)
      KGM7=NN(I,1)-NN(I-1,1)
      KGM8=NN(I,2)-NN(I-1,2)
      KGMMX=MAX0(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7,KGM8)
      KGMMN=MIN0(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7.KGM8)
      IF(KGMMX.EQ.KGMMN)   KGM=KGMMX
      end if

      KGP=0
      if(i.lt.ntoe.and.
     .IDM(I).EQ.IDM(I+1).AND.MAXES(I).EQ.MAXES(I+1).AND.IOP.EQ.IOP
     .     .AND.TZ(I).EQ.TZ(I+1).AND.MAT(I).EQ.MAT(I+1)) then
      KGP1=NN(I+1,7)-NN(I,7)
      KGP2=NN(I+1,8)-NN(I,8)
      KGP3=NN(I+1,5)-NN(I,5)
      KGP4=NN(I+1,6)-NN(I,6)
      KGP5=NN(I+1,3)-NN(I,3)
      KGP6=NN(I+1,4)-NN(I,4)
      KGP7=NN(I+1,1)-NN(I,1)
      KGP8=NN(I+1,2)-NN(I,2)
      KGPMX=MAX0(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
      KGPMN=MIN0(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
      IF(KGPMX.EQ.KGPMN)   KGP=KGPMX
      end if
      ISKIP=0
cc    ISKIP=1
      kgz=0
```

```
      if(iskipo.eq.0.and.kgp.gt.0) kgz=kgp
      I1=I
c     WRITE(9,*) I,ISKIPO,ISKIP,KGM,KGP
      IF(ISKIP.EQ.0)THEN
      WRITE(9,13025)I1,IDM(I1),MAXES(I1),IOP,TZ(I1),kgz
     .,MAT(I1),(NN(I1,JX),JX=1,NTYPE)
       END IF
13025 FORMAT(I5,I15,2I5,F10.0,I5,10X,I5/16I5/8I5)
      ISKIPO=ISKIP
      KGMO=KGM
      END DO
1302  FORMAT(I5,I15,2I5,F10.0,15X,I5/8I5)

      NBMN=0
      DO J=1,NTBF
      NBM=99999
      DO I=1,NTBF
      IF(NBF(I).LT.NBM.AND.NBF(I).GT.NBMN) THEN
      NBM=NBF(I)
      IM=I
      END IF
      END DO
      ID1(J)=Im
      NBMN=NBM
      END DO

      DO J=1,NTBF
      I=ID1(J)
      WRITE(9,1303) NBF(I),FX(I),FY(I),FZ(I)
      END DO
 1303 FORMAT(I5,4X,'1',3F10.4)

      WRITE(9,1305)
 1305 FORMAT(/8X,'1.')

      WRITE (9,1306)
1306  FORMAT(1X,'0'/1X,'0'/1X,'0 0 0'//1X,'9999 9999 0')
      WRITE (10,*) '                                       '
      WRITE (10,*) ' ....For Crack Simulation...'
      WRITE (10,*) ' ORIG. NODES  DOUBLE NODES '
      WRITE (10,*) ' ---------------------------'
      ntono=nton-ntd-nsd
      write (9,8128) ntd
      do 850 i=1,ntono
      if (mip(i,2).ne.0) then
      do 860 j=1,nsd
      if (i.eq.nons(j)) write(10,8126) I,mip(i,1),mip(i,2)
      if (i.eq.nons(j)) go to 850
860   continue
      write(9,8125) mip(i,1),mip(i,2),I
      end if
850   continue
8125  format(2i5,7h  1 1 1, '      !   ',i5)
8126  format(2X,I5,3X,2I5)
```

```
c      do i=1,ntono
c      if(mip(i,2).ne.0) then
c--   writing in the order they are given in the input file----
      do i=1,ntd
      ii=nond(i)
      do j=1,3
      write(9,8128) mip(ii,1),mip(ii,2),j,ii
      end do
8128  format(2i5,i3,5X,'    !    ',I5)
c      end if
      end do

      CLOSE (UNIT=9)
      STOP
     END
c
c%=================================================================
      subroutine split (NTON,ntd,idir,nond,mip,x,y,z,T,
     .ntype,nos,NN,ntoe)
      dimension nond(1),mip(4000,2),iface(9),jface(9),nos(1),T(1)
     .,xd(400),yd(400),zd(400),x(1),y(1),z(1),nn(3000,21)
      NFE=4
      IF (NTYPE.EQ.21) NFE=9
C     STORE COORD. OF THE NODES TO BE DOUBLED
C-  CORRECTIONS FOR DOUBLE NODES IN x or y or z -direction--

      DO 819 I=1,4
      IFACE(I)=I+4
819   JFACE(I)=I
      IFACE(5)=13
      IFACE(6)=14
      IFACE(7)=27
      IFACE(8)=16
      IFACE(9)=15
      JFACE(5)=9
      JFACE(6)=10
      JFACE(7)=26
      JFACE(8)=12
      JFACE(9)=11
      NTELR=NOS(1)*NOS(2)  !!NTELR--TOTAL NUMBER OF ELEM. IN AN ELE-LAYER
      IF (IDIR.LT.1.OR.IDIR.GT.3) IDIR=3
      IF (IDIR-2) 901,902,903
901   IFACE(1)=2
      IFACE(2)=3
      IFACE(3)=6
      IFACE(4)=7
      JFACE(1)=1
      JFACE(2)=4
      JFACE(3)=5
      JFACE(4)=8
      IFACE(5)=10
      IFACE(6)=18
      IFACE(7)=23
      IFACE(8)=19
```

```
      IFACE(9)=14
      JFACE(5)=12
      JFACE(6)=17
      JFACE(7)=22
      JFACE(8)=20
      JFACE(9)=16
      NTELR=1
      GO TO 903
902   IFACE(1)=3
      IFACE(2)=4
      JFACE(1)=2
      JFACE(2)=1
      JFACE(3)=6
      JFACE(4)=5
      IFACE(5)=11
      IFACE(6)=19
      IFACE(7)=25
      IFACE(8)=20
C      IFACE(9)=15
C      JFACE(5)=9
      JFACE(6)=18
      JFACE(7)=24
      JFACE(8)=17
      JFACE(9)=13
      NTELR=NOS(1)
903   CONTINUE

      DO 701 I=1,NTD-1
      ICHANGE-0
      DO 702 J=1,NTD-I
      JJ=J+1
      IF (NOND(J).LT.NOND(JJ)) GO TO 702
      ICHANGE=1
      AA=NOND(J)
      NOND(J)=NOND(JJ)
      NOND(JJ)=AA
702   CONTINUE
      IF (ICHANGE.EQ.0) GO TO 703
701   CONTINUE

703   CONTINUE
      DO I=1,NTD
      XD(I)=X(NOND(I))
      YD(I)=Y(NOND(I))
      ZD(I)=Z(NOND(I))
      END DO
C-- CORRECTIONS DOUB. NODES. ENDS.--

C   FIND KND (THE # OF THE NODE CURRENTLY TO BE DOUBLED)
      DO 830 I=1,NTD
      DO 820 J=NOND(I),NTON
      DIF=SQRT((X(J)-XD(I))**2+(Y(J)-YD(I))**2+
     .(Z(J)-ZD(I))**2)
C      IF(X(J).NE.XD(I)) GO TO 820
```

```
C        IF(Y(J).NE.YD(I)) GO TO 820
C        IF(Z(J).NE.ZD(I)) GO TO 820
         IF (DIF.GT.1.0E-13) GO TO 820
         KND=J
         GO TO 825        ! GET OUT OF J LOOP
    820  CONTINUE
    825  CONTINUE
C        BEGIN CHANGING NODE NUMBERS & COORD.
C        CHANGE NODE NUMBERS
         DO J=1,NTOE
         DO K=1,ntype
         IF (NN(J,K).GT.KND) THEN
         NN(J,K)=NN(J,K)+1
         END IF
         END DO
         END DO

C-- substitution FOR DOUB. NODES x or y or z dir.--

         DO K=1,NTOE
         DO M=1,NFF
         MM=JFACE(M)
         IF (NN(K,MM).EQ.KND) THEN
         MJ=IFACE(M)
         NN(K+NTELR,MJ)=KND+1
         END IF
         END DO
         END DO

c--      corrections for doub. nodes ends.--

C        CHANGING COORD.
         DO J=NTON,KND,-1
         X(J+1)=X(J)
         Y(J+1)=Y(J)
         Z(J+1)=Z(J)
         T(J+1)=T(J)
         END DO
         NTON=NTON+1
    830  CONTINUE
         return
         end
c*====================================================
         subroutine delete(nonx,nony,nonz,ntd,nond)
         dimension nond(1)
         ixx=nonz
         iyy=nonx*nonz
         IFNO=0
         DO 422 I=1,NONZ
         DO 422 J=1,NONY
         DO 422 K=1,NONX
         IF (MOD(I,2).EQ.0) GO TO 411
         IF (MOD(J,2).NE.0.OR.MOD(K,2).NE.0) GO TO 422
c        I-odd, j,k-even..........
```

```
      go to 421
 411  if(mod((j+k),2).eq.0) go to 422
 421  i3=iyy*(j-1)+ixx*(k-1)+1
      do 425 l=1,ntd
      if (i3.ne.nond(l)) go to 425
      ifno=ifno+1
      do 426 m=l+1,ntd
 426  nond(m-1)=nond(m)
      go to 427
 425  continue
      go to 422
 427  ntd=ntd-1
 422  continue
C      type *,ntd
C      type *,(nond(i),i=1,ntd)
      return
      end

c&===========================================================================
        SUBROUTINE MESHPL (XX,YY,NONX,NONY,NONZ,MIP,IPLOT,ntype)
c-------------------------------------------------
c--     Program 'MESH.FOR' to be used with 'PREPRO214.FOR'
c--     This routine plots the mesh in xy-direction..
c--     This routine is based on HP-GL language. The
c--     plot cannot be displayed on TTY. Before using
c--     this routine ASSIGN a HP plotter to FOR090
c--     IPLOT = 1 original nodes
c--     IPLOT = 2 after double nodes
C-------------------------------------------------
        LOGICAL*1 ETX,ESC
        CHARACTER *1 JUNK,MESH
        DATA ETX/'3/,ESC/'33/
        DIMENSION X(4000),Y(4000),XX(1),YY(1)
     .,mip(4000,2)
        NTOT=NONX*NONY*NONZ
        DO 2 I=1,NTOT
        X(I)=XX(I)
 2      Y(I)=YY(I)

        IX=NONZ
        IY=NONX*NONZ

C--     SCALING THE COORDINATES-----------

        XMIN=5000.
        XMAX=-5000.
        YMIN=5000.
        YMAX=-5000.
        DO 50 I=1,NTOT
        IF (X(I).GT.XMAX) XMAX=X(I)
        IF (X(I).LT.XMIN) XMIN=X(I)
        IF (Y(I).GT.YMAX) YMAX=Y(I)
 50     IF (Y(I).LT.YMIN) YMIN=Y(I)
        WRITE (5,*) 'TYPE LEVEL no. to plotted..'
```

```
        READ (5,*) LEVEL
        N=LEVEL
        XLL=XMIN
        YLL=YMIN
        XMM=XMAX
        YMM=YMAX
        XL=XMM-XLL
        YL=YMM-YLL
        WRITE (5,*) ' DO YOU WANT TO GIVE X,Y LIMITS..? (Y/N)'
        READ (5,33) MESH
        IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') THEN
        WRITE (5,*)' ENTER X- LIMITS'
        READ (5,*) XLL,XMM
        WRITE (5,*)' ENTER Y- LIMITS'
        READ (5,*) YLL,YMM
        XL=XMM-XLL
        YL=YMM-YLL
        END IF
        XO=650.0
        YO=1246.0
        Xc= 8800.0/XL
        Yc=6000.0/YL
        SC=xc
        IF (XC*yl.gt.6000.0) SC=yc

        IF (XL.LT.YL) THEN
        XO=1246.0
        YO=650.0
        Xc= 6000.0/XL
        Yc=8800.0/YL
        SC=xc
        IF (XC*yl.gt.8800.0) SC=yc
        END IF
C        type *,'xl,yl',xl,yl
        DO 1 I=1,NTOT
        X(I)=(X(I)-XLL)*SC+XO
1       Y(I)=(Y(I)-YLL)*SC+YO

C        type *,x(1),x(ntot)
C        type *,y(1),y(ntot)
888     WRITE (90,9999) ESC,ESC,ESC,ESC
 9999   FORMAT (' ',A1,'.(',/,' ',A1,'.@;0:',/,' ',A1,'.I40;;17:',/,' ',
      . A1,'.N;19:',/,' IN;DF;',/)
        WRITE (90,*),'SP1;'
        WRITE (90,*),' VS15.0;'
        IF (XL.LT.YL) THEN
        WRITE (90,*) ' RO90;'
        WRITE (90,*) ' IP;IW;'
        END IF

        XMM=(XMM-XLL)*SC+XO
        YMM=(YMM-YLL)*SC+YO
        XLL=XO
        YLL=YO
```

```
      J1=level
      J2=(nony-1)*iy+j1
C      DO 111 I=1,NONY
C111   TYPE *,Y(I),YLL
C      CALL LIMIT (Y,YLL,j1,j2,iy,J1)
C      CALL LIMIT (Y,YMM,J1,j2,iy,J2)
C      J2=IF2(Y,YMM,J1,NONY,1)
C      TYPE *,'20 LOOP',J1,J2
      DO 20 J=J1,J2,iy
      NX1=j
      NX2=nx1+IY-ix
      CALL LIMIT (X,XLL,NX1,NX2,IX,NX1)
      CALL LIMIT (X,XMM,NX1,NX2,IX,NX2)
C      NX1=IF1(X,XLL,NX1,NX2,IX)
C      NX2=IF2(X,XMM,NX1,NX2,IX)
C       TYPE *,' 10 LOOP ',NX1,NX2
C      TYPE *,X(NX1),Y(NX1),X(NX2),Y(NX2)
      DO 10 I=NX1,NX2,IX
      IF (i.eq.nx1) WRITE (90,101), X(I),Y(I)
      WRITE (90,104), X(I),Y(I)
104   FORMAT (' PD',2F11.3,';')
10    continue
20    WRITE (90,*) ' PU;'

      I1=level
      I2=(NONX-1)*ix+i1
C      CALL LIMIT (X,XLL,i1,i2,IX,i1)
C      CALL LIMIT (X,XMM,i1,i2,IX,i2)
C      I1=IF1(X,XLL,1,NONX,1)
C      I2=IF2(X,XMM,I1,NONX,1)
C      TYPE *,' 30 LOOP',I1,I2
      DO 30 I=I1,I2,IX
      NY1=i
      NY2=NY1+(nony-1)*IY
C      CALL LIMIT (Y,YLL,NY1,NY2,iy,NY1)
C      CALL LIMIT (Y,YMM,NY1,NY2,iy,NY2)
C      NY1=IF1(Y,YLL,NY1,NY2,IY)
C      NY2=IF2(Y,YMM,NY1,NY2,IY)
C      TYPE *, ' 40 LOOP ',NY1,NY2
      DO 40 J=NY1,NY2,IY
      IF (J.EQ.NY1) WRITE (90,101), X(J),Y(J)
      WRITE (90,102), X(J),Y(J)
40    CONTINUE
101   FORMAT (' PU;PA',2F11.3,';')
102   FORMAT (' PD',2F11.3,';')
30    WRITE (90,*) ' PU;'

      WRITE (5,*) 'DO YOU NEED NODE NOS. (Y/N)..'
      READ (5,33) JUNK
33    FORMAT (A)
      IF (JUNK.EQ.'Y'.OR.JUNK.EQ.'y') GO TO 44
      GO TO 66
!      WRITE (5,*) '..ENTER SSX,SSY...'
```

```
!      READ (5,*) SSX,SSY
44      SSX=0.1
       SSY=0.15
!--    WRITING THE NODE NUMBERS---------
       WRITE (90,*)'SI0.1,0.15;'
       J1=level
       J2=(nony-1)*iy+j1
C       CALL LIMIT (Y,YLL,j1,j2,iy,J1)
C       CALL LIMIT (Y,YMM,J1,j2,iy,J2)
       DO 100 J=J1,J2,IY
       NX1=j
       NX2=nx1+IY-ix
C       CALL LIMIT (X,XLL,NX1,NX2,IX,NX1)
C       CALL LIMIT (X,XMM,NX1,NX2,IX,NX2)

!       TYPE *,(KK,KK=NX1,NX2,IX)
       DO 100 IN=NX1,NX2,IX
       NXX=(IN-NX1)/IX+1        !skipping the face nos.
       IF (MOD(J,2).EQ.0.AND.MOD(NXX,2).EQ.0.and.
     . ntype.eq.21)GO TO 100
       I=IN
       IF (IPLOT.EQ.2) I=MIP(IN,1)
       SPL=-0.8
       XXX=X(IN)
       YYY=Y(IN)
       IF (XXX.LT.XLL.OR.YYY.LT.YLL) GO TO 100
       IF (XXX.GT.XMM.OR.YYY.GT.YMM) GO TO 100
       CALL SYMB (XXX,YYY,SPL,I)
       IF (IPLOT.EQ.1) GO TO 100
       I=MIP(IN,2)
       SPL=0.2
       IF (I.NE.0) CALL SYMB (XXX,YYY,SPL,I)
100     WRITE (90,*) ' PU;'
66     WRITE (90,*),'SP0;'
       WRITE (5,*) 'TYPE another LEVEL no. to plotted..'
       READ (5,*) LEVEL
       if (level.ne.0) go to 888
       RETURN
       END

       SUBROUTINE SYMB (X,Y,SPL,I)
       LOGICAL*1 ETX,ESC
       DATA ETX/'3/,ESC/'33/
       NC=1
       IF (I.GT.9) NC=2
       IF (I.GT.99) NC=3
       IF (I.GT.999) NC=4
       IF (I.GT.9999) NC=5
       IF (I.GT.99999) NC=6
       SPC=-(0.33+0.5*(NC-1))+2.
       WRITE (90,101), X,Y
       IF (NC.EQ.1) WRITE (90,201) SPC,SPL,I,ETX
       IF (NC.EQ.2) WRITE (90,202) SPC,SPL,I,ETX
       IF (NC.EQ.3) WRITE (90,203) SPC,SPL,I,ETX
```

```
      IF (NC.EQ.4) WRITE (90,204) SPC,SPL,I,ETX
      IF (NC.EQ.5) WRITE (90,205) SPC,SPL,I,ETX
      IF (NC.EQ.6) WRITE (90,206) SPC,SPL,I,ETX
101   FORMAT (' PU;PA',2F11.3,';')
201   FORMAT (' CP',2F6.2,';LB',I1,A2)
202   FORMAT (' CP',2F6.2,';LB',I2,A2)
203   FORMAT (' CP',2F6.2,';LB',I3,A2)
204   FORMAT (' CP',2F6.2,';LB',I4,A2)
205   FORMAT (' CP',2F6.2,';LB',I5,A2)
206   FORMAT (' CP',2F6.2,';LB',I6,A2)
      RETURN
      END

      SUBROUTINE LIMIT (A,AL,I1,I2,IN,IF1)
c--   to find the lower limit of do loop...
      DIMENSION A(1)
      DO 10 I=I1,I2,IN
C      TYPE *,A(I),AL
      IF (A(I).GE.(AL-1.0E-03)) THEN
      IF1=I
      RETURN
      END IF
10    CONTINUE
      END

      INTEGER FUNCTION IF2(A,AL,I1,I2,IN)
c--   to find the lower limit of do loop...
      DIMENSION A(1)
      IF2=I2
      DO 10 I=I1,I2,IN
C      type *,a(i),al
      IF (A(I).GE.(AL-1.0E-03)) THEN
      IF1=I
      RETURN
      END IF
10    CONTINUE
      END

c&=========================================================
        SUBROUTINE HOLE(rx,NONX,NONY,NONZ,XX,YY,ZZ,X,Y,Z)
C   GENERATING POLAR COORDINATES AND POLAR MESH--
        DIMENSION XX(1),YY(1),ZZ(1),X(1),Y(1),R(100),NC(4)
     . ,Z(1)
c        TYPE *,(XX(I),I=1,NONX)
c        TYPE *,(YY(I),I=1,NONY)
c        TYPE *,(ZZ(I),I=1,NONZ)
        pi=3.14159265
        EPS=1.0E-05

        IX=NONZ
        IY=NONX*NONZ
        DO 31 I=1,NONX
        rad=rx
        if (xx(i).ne.0.0)rad=sqrt(xx(i)**2+rx**2)
```

```
      DO 31 J=1,NONY

      xxx=xx(i)
      if (yy(j).le.(rx-eps)) then
      xxx=rad
      if (yy(j).ne.0.)xxx=sqrt(rad**2-yy(j)**2)
      nbk=(j-1)*iy+(i-2)*ix+1
      TYPE *,NBK
      if ((xxx-x(nbk)).lt.(xx(2)-xx(1)))xxx=x(nbk)+xx(2)-xx(1)
      end if

      nxy=(J-1)*IY+(I-1)*IX
      DO 31 K=1,NONZ
      N=nxy+K
      X(N)=xxx
      IF (I.EQ.NONX) X(N)=XX(NONX)
      Y(N)=YY(J)
      Z(N)=ZZ(K)
 31   continue
C--   correcting for center nodes for 21 node element....

      DO 20 K=1,NONZ
      DO 20 J=1,NONY
      DO 20 I=1,NONX
      IF (MOD(I,2).NE.0) GO TO 20
      N=(J-1)*IY+(I-1)*IX+K
      X(N)=(X(N-IX)+X(N+IX))/2.0
 20   CONTINUE
      do 30 j=1,nony
      do 30 i=1,nonx
      n=(j-1)*iy+(i-1)*ix+1
      type *,n,x(n),y(n)
 30   CONTINUE
      RETURN
      END
```

APPENDIX - B

LISTING OF THE MAIN CODE 'KSAP II'

```
C  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  *
C
C                       KSAP II
C                SIMPLIFIED VERSION OF SAP4 FOR
C                  USING ELEMENT TYPE 8 ONLY
C
C                       September 1987
C
C
C
C  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  *
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 T,TT
      COMMON /JUNK / HED(12),JUK(406)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/  QQQ(2846)
      COMMON /DYN/   IDUS(11),NDYN
      COMMON /TAPES/ NQQ(6)
      COMMON /EXTRA/ MODEX,NT8,N10SV,NT10,KEQB,NUMEL,T(10)
      COMMON /SOL/   NBLOCK,NEQB,LL,NF,IDUM,NEIG,NAD,NVV,ANORM,NEO
C
C     PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS ...
C
      COMMON A(650001)       !CHANGE MTOT ALSO
C
C --  OPEN SCRATCH FILES
c      OPEN (UNIT=1,STATUS='UNKNOWN',FORM='UNFORMATTED')
*-----------------------------------------------------------------------
      open(unit=1,FILE='SCR:[ASW.EMANI]',status='scratch',
     .form='unformatted')
!       open(unit=2,status='scratch',form='unformatted')          !dra0
      open (unit=2,file='SCR:[ASW.EMANI]',status='new',blocksize=4800,
     .form='unformatted')                                          !msa0:
!      open(unit=3,status='scratch',form='unformatted')           !dra0:
      open (unit=3,file='UTL:[ASW.EMANI]',status='new',blocksize=4800,
     .form='unformatted')                                          !msa0:
!      open (unit=3,file='MSA0:[ESR]',status='new',blocksize=4800,
!     .form='unformatted')                                         !msa0:
      OPEN (UNIT=4,FILE='SCR:[ASW.EMANI]',STATUS='NEW',BLOCKSIZE=4800,
     .FORM='UNFORMATTED')                                          !dma0:
!      open(unit=55,status='UNKNOWN',form='unformatted')          !dra0
      OPEN (UNIT=55,FILE='SCR:[ASW.EMANI]',STATUS='NEW',BLOCKSIZE=4800,
     .FORM='UNFORMATTED')                                          !DMA0:
      open(unit=8,FILE='SCR:[ASW.EMANI]',status='scratch',
     .FORM='UNFORMATTED')                                          !dma0:
      open(unit=9,FILE='SCR:[ASW.EMANI]',status='scratch',
     .FORM='UNFORMATTED')                                          !dma0:
C      OPEN (UNIT=15,STATUS='SCRATCH',FORM='UNFORMATTED')         !DRA0:
      open(unit=16,FILE='SCR:[ASW.EMANI]',status='scratch',
     .FORM='UNFORMATTED')                                          !dma0:
      open(unit=18,FILE='SCR:[ASW.EMANI]',status='scratch',
     .FORM='UNFORMATTED')                                          !dma0:
      open (unit=19,file='workdone.wok',status='new')             !dra0:
```

```
      open (unit=33,file='SCR:[ASW.EMANI]DISP.dat',status='new')
      open (unit=34,file='ksapout.dat',status='new')
      OPEN (UNIT=15,FILE='SCR:[ASW.EMANI]',STATUS='NEW',BLOCKSIZE=4800,
     .FORM='UNFORMATTED')

*-----------------------------------------------------------------------

!    THE following should be 1 less than A() dimension
      MTOT=  650000         !  300000

C     USE THE IBM FORTRAN EXTENDED ERROR HANDLING FACILITY TO
C     ELIMINATE PRINTOUT OF UNDERFLOW ERROR MESSAGE (ERROR NUMBER 208)
C
C      CALL ERRSET (208,256,-1,1)
C
C
      CALL STIME
C
      NT8 = 8
      REWIND NT8
      NT10= 10
      REWIND NT10
      N1=1
C
C     P R O G R A M   C O N T R O L   D A T A
C
    5 CALL TTIME(T(1))
      READ (5,100,END=990) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,
     1                     KEQB,N1OSV,NDOF
      IF(MODEX.GT.0) MODEX = 1
      IF (NUMNP.EQ.0) STOP
      WRITE (33,200) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,KEQB,N1OSV
      WRITE (19,299) HED  !WORKDONE.WOK FILE TILTE.....
      WRITE (34,299) HED  !KSAPOUT.DAT (STRESSES) FILE TILTE.....
      IF(KEQB.LT.2) KEQB = 99999
      IF (NDYN.NE.0) LL=1
      IF(LL.GE.1) GO TO 10
      WRITE (33,300)
      STOP
C***  DATA PORTHOLE SAVE
   10 IF(MODEX.EQ.1)
     *WRITE (NT8)     HED,NUMNP,NELTYP,LL,NF,NDYN
C
      KDYN = IABS(NDYN) +1
      IF(KDYN.LE.5) GO TO 14
      WRITE (33,310) NDYN
      STOP
C
C     RE-START MODE ACTIVATED IF  NDYN.EQ.-2  OR  NDYN.EQ.-3
C
   14 IF(NDYN.LT.0) GO TO 20
C
C     I N P U T   J O I N T   D A T A
C
```

```
      N2=N1+6*NUMNP
      N3=N2+NUMNP
      N4=N3+NUMNP
      N5=N4+NUMNP
      N6=N5+NUMNP
      IF(N6.GT.MTOT) CALL ERROR(N6-MTOT)
C
      CALL INPUTJ(A(N1),A(N2),A(N3),A(N4),A(N5).NUMNP,NEQ)
C
C     F O R M   E L E M E N T   S T I F F N E S S E S
C
      CALL TTIME(T(2))
C
      MBAND=0
      NUMEL=0
      REWIND 1
      REWIND 2
C
      DO 900 M=1,NELTYP
      READ  (5.1001) NPAR
C***  DATA PORTHOLE SAVE
      IF(MODEX.EQ.1) WRITE (NT8) NPAR
      WRITE (1) NPAR
      NUMEL=NUMEL+NPAR(2)
      MTYPE=NPAR(1)
C
      CALL ELTYPE(MTYPE)
C
  900 CONTINUE
C
C     D E T E R M I N E   B L O C K S I Z E
C
C     ADDSTF
C
      LL1=LL+NDOF       !IN the following LL is replaced with LL1
      NEQB=(MTOT - 4*LL)/(MBAND + LL1 + 1)/2   !modified with ndof
C
C     OVER-RIDE THE SYSTEM MATRIX BLOCKSIZE WITH THE INPUT (NON-ZERO)
C     VALUE, KEQB.
C     THIS OVER-RIDE ENTRY IS TO ALLOW PROGRAM CHECKING OF MULTI-
C     BLOCK ALGORITHMS WITH WHAT WOULD NORMALLY BE ONE BLOCK DATA.
C
      IF(KEQB.LT.NEQB) NEQB = KEQB
C
      GO TO  (690,700,700,700,730), KDYN
C
C     STATIC SOLUTION
C
  690 CONTINUE
      NEQB1=(MTOT - MBAND)/(2*(MBAND+LL1) + 1)
      NEQB2=(MTOT - MBAND - LL1 *(MBAND-2))/(3*LL1 + MBAND + 1)
      IF (NEQB1.LT.NEQB) NEQB=NEQB1
      IF (NEQB2.LT.NEQB) NEQB=NEQB2
      NBLOCK = (NEQ-1)/NEQB +1
```

```
      IF(NEQB.GT.NEQ) NEQB=NEQ
      GO TO 790
C
C     EIGENSOLUTION
C
C        1.  DETERMINANT SEARCH ALGORITHM
C
  700 IF (NEQB.LT.NEQ) GO TO 710
      NIM=3
      NC=NE + NIM
      NVM=6
      NCA=NEQ*MAX0(MBAND,NC)
      NTOT=NCA + 4*NEQ + 3*NVM*NEQ + 5*NC
      NEIG=0
      IF(NTOT.LE.MTOT) GO TO 720
C
C        2.  SUBSPACE ITERATION ALGORITHM
C
  710 NV=MIN0(2*NE,NE+8)
      IF (NAD.NE.0) NV=NAD
      NEQB1=(MTOT - MBAND)/(2*MBAND + 1)
      NEQB2=(MTOT - MBAND - 2*NV - NV*(MBAND-2))/(3*NV + MBAND + 1)
      NEQB3=(MTOT - 3*NV*NV - 3*NV)/(2*NV + 1)
      NEQB4=(MTOT - 6*NV)/(1 + MBAND)
      IF (NEQB1.LT.NEQB) NEQB=NEQB1
      IF (NEQB2.LT.NEQB) NEQB=NEQB2
      IF (NEQB3.LT.NEQB) NEQB=NEQB3
      IF (NEQB4.LT.NEQB) NEQB=NEQB4
      NEIG=1
C
  720 CONTINUE
      NBLOCK = (NEQ-1)/NEQB +1
      IF (NEQB.GE.NEQ) NEQB=NEQ
C
C     HISTORY OR SPECTRUM ANALYSIS
C
      KREM = 1000
      NTOT = NBLOCK*NEQB*NE + KREM
      IF(MTOT.LT.NTOT)
     *WRITE (33,320)
      GO TO 790
C
C     STEP-BY-STEP DIRECT INTEGRATION
C
  730 CONTINUE
C     DISPLACEMENT COMPONENTS FOR DIRECT OUTPUT (*NSD*)
      NN2 = NEQ
C     DISPLACEMENT COMPONENTS REQUIRED FOR RECOVERY OF ALL OF THE
C     REQUESTED ELEMENT STRESS COMPONENTS (*NSS*)
      NN3 = NEQ
C
C        1.  DECOMPOSITION
C
      NEQB1 = (MTOT-NN2-NN3-NEQ-MBAND)/(2*MBAND+1)
```

```
C
C          2.  TIME INTEGRATION PHASE
C
      mcal= MBAND+2*(NN2+NN3)+5*NEQ +(2*MBAND+1)


      write (33,555) mcal
  555 format(//5x,' Minimum dimension, MTOT, required for array A( )
     . = ',I8/5X,50(1H+)//)
      if(mtot.le.mcal) STOP        ! Abnormal stop as dim. of A is insufficient

      NEQB2 = (MTOT-MBAND-2*(NN2+NN3)-5*NEQ)/(MBAND+1)
C
      IF(NEQB1.LT.NEQB) NEQB = NEQB1
      IF(NEQB2.LT.NEQB) NEQB = NEQB2
      IF(NEQB.GT.NEQ)   NEQB = NEQ
      NBLOCK = (NEQ-1)/NEQB +1
C
C          3.  INPUT PHASE
C
C     NUMBER OF TIME FUNCTIONS (*NFN*)
      NN2 = 10
C     MAXIMUM NUMBER OF FUNCTION DEFINITION POINTS (*MXLP*)
      NN3 = 40
C
      NN4 = 6*NUMNP + 2*NN2*NEQ
      IF(NN4.GT.MTOT)
     *WRITE (33,320)
      NN4 = NEQ*2*(NN2+1) + NN2*(1+2*NN3)
      IF(NN4.GT.MTOT)
     *WRITE (33,320)
C
  790 CONTINUE
C
C     I N P U T   N O D A L   L O A D S
C
      N3=N2+NEQB*LL
      N4=N3+6*LL
      WRITE (33,201) NEQ,MBAND,NEQB,NBLOCK
C
      CALL TTIME(T(3))
C
      CALL INL(A(N1),A(N2),A(N3),A(N4),NUMNP,NEQB,LL)
C
      CALL TTIME(T(4))
C
C     F O R M   T O T A L   S T I F F N E S S
C
      NE2B=2*NEQB
      N2=N1+NEQB*MBAND
      N3=N2+NEQB*LL
      N4=N3+4*LL
      NN2=N1+NE2B*MBAND
      NN3=NN2+NE2B*LL
```

```
      NN4=NN3+4ALL
C
      CALL ADDSTF (A(N1),A(NN2),A(NN3),A(NN4),NUMEL,NBLOCK,NE2B,LL,MBAND
     1,ANORM,NVV)
C
      CALL TTIME(T(5))
C
C     S O L U T I O N   P H A S E
C
   20 GO TO (30,40,50,60,70), KDYN
C
C     STATIC SOLUTION
C
   30 IF(MODEX.EQ.0) GO TO 32
      DO 31 I=6,10
   31 T(I) = T(5)
      GO TO 90
C
   32 CALL SOLEQ
      CALL TTIME(T(6))
      DO 33 I=7,10
   33 T(I) = T(6)
      GO TO 90
C
C     EIGENVALUE EXTRACTION
C
   40 T(6) = T(5)
C     CALL SOLEIG
      CALL TTIME(T(7))
      T(8) = T(7)
      T(9) = T(7)
      T(10)= T(7)
      GO TO 90
C
C     FORCED DYNAMIC RESPONSE ANALYSIS
C
   50 T(6) = T(5)
      IF(NDYN.LT.0) GO TO 52
C     CALL SOLEIG
      CALL TTIME (T(7))
      GO TO 54
   52 DO 53 I=1,6
   53 T(I+1)=T(I)
      REWIND 2
      READ (2) NEQ,NBLOCK,NEQB,MBAND,N1,NE,(QQQ(I),I=1,NE)
      REWIND 55
      IMAX=NEQBANF
      READ (55) (A(I),I=1,NE)
      DO 56 L=1,NBLOCK
   56 READ (55) (A(I),I=1,IMAX)
   54 CONTINUE
C  54 CALL HISTRY
      CALL TTIME (T(8))
      T(9) = T(8)
```

```
      T(10)= T(8)
      GO TO 90
C
C     RESPONSE SPECTRUM ANALYSIS
C
   60 T(6) = T(5)
      IF(NDYN.LT.0) GO TO 62
C     CALL SOLEIG
      CALL TTIME (T(7))
      T(8) = T(7)
      GO TO 64
   62 DO 63 I=1,7
   63 T(I+1)=T(I)
      REWIND 2
      READ (2) NEQ,NBLOCK,NEQB,MBAND,N1,NF
      REWIND 55
      IMAX=NEQB*NF
      READ (55) (A(I),I=1,NF)
      DO 66 L=1,NBLOCK
   66 READ (55) (A(I),I=1,IMAX)
   64 CONTINUE
C  64 CALL RESPEC
      CALL TTIME (T(9))
      T(10)= T(9)
      GO TO 90
C
C     STEP-BY-STEP (DIRECT INTEGRATION) ANALYSIS
C
   70 DO 71 I=6,9
   71 T(I) = T(5)
C     CALL STEP
      CALL TTIME(T(10))
C
C     COMPUTE AND PRINT OVERALL TIME LOG
C
   90 TT = 0.0
      DO 95 I=1,9
      T(I) = T(I+1)-T(I)
      TT = TT + T(I)
   95 CONTINUE
C
      WRITE (33,203) (T(K),K=1,9),TT
C
      GO TO 5
  990 CONTINUE
C -- CLOSE ALL SCRATCH FILES
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      CLOSE (UNIT=55)
      CLOSE (UNIT=8)
      CLOSE (UNIT=9)
      close (unit=15)
```

```
      close (unit=16)
      close (unit=18)
      close (unit=19)
C
      STOP
C
  100 FORMAT (12A6/10I5)
299   FORMAT(/2X,12A6/)
200   FORMAT(1H1,12A6///
     1  38H C O N T R O L   I N F O R M A T I O N, // 4X,
     2  27H NUMBER OF NODAL POINTS   =, I5 / 4X,
     3  27H NUMBER OF ELEMENT TYPES  =, I5 / 4X,
     4  27H NUMBER OF LOAD CASES     =, I5 / 4X,
     5  27H NUMBER OF FREQUENCIES    =, I5 / 4X,
     6  27H ANALYSIS CODE (NDYN)     =, I5 / 4X,
     7  16H   EQ.0,  STATIC,               / 4X,
     8  26H   EQ.1,  MODAL EXTRACTION,     / 4X,
     9  25H   EQ.2,  FORCED RESPONSE,      / 4X,
     A  27H   EQ.3,  RESPONSE SPECTRUM,    / 4X,
     *  28H   EQ.4,  DIRECT INTEGRATION,   / 4X,
     B  27H SOLUTION MODE (MODEX)    =, I5 / 4X,
     C  19H   EQ.0,  EXECUTION,            / 4X,
     D  20H   EQ.1,  DATA CHECK,           / 4X,
     E  19H NUMBER OF SUBSPACE,            / 4X,
     F  27H ITERATION VECTORS (NAD)  =, I5 / 4X,
     G  27H EQUATIONS PER BLOCK      =, I5 / 4X,
     H  27H TAPE10 SAVE FLAG (N10SV) =, I5 / 4X)
  201 FORMAT (38H1E Q U A T I O N   P A R A M E T E R S, //
     *        34H  TOTAL NUMBER OF EQUATIONS      =,I5,
     1       /34H  BANDWIDTH                      =,I5,
     2       /34H  NUMBER OF EQUATIONS IN A BLOCK =,I5,
     3       /34H  NUMBER OF BLOCKS               =,I5)
  203 FORMAT (1H1,31HO V E R A L L   T I M E   L O G, //
     1 5X,30HNODAL POINT INPUT              =, F8.2 /
     2 5X,30HELEMENT STIFFNESS FORMATION    =, F8.2 /
     3 5X,30HNODAL LOAD INPUT               =, F8.2 /
     4 5X,30HTOTAL STIFFNESS FORMATION      =, F8.2 /
     5 5X,30HSTATIC ANALYSIS                =, F8.2 /
     6 5X,30HEIGENVALUE EXTRACTION          =, F8.2 /
     7 5X,30HFORCED RESPONSE ANALYSIS       =, F8.2 /
     8 5X,30HRESPONSE SPECTRUM ANALYSIS     =, F8.2 /
     * 5X,30HSTEP-BY-STEP INTEGRATION       =, F8.2 //
     9 5X,30HTOTAL SOLUTION TIME            =, F8.2 /)
C
  300 FORMAT (// 48H ** ERROR.  (AT LEAST ONE LOAD CASE IS REQUIRED)  )
  310 FORMAT (// 33H ** ERROR.  ANALYSIS CODE (NDYN =,I3,9H) IS BAD.  )
  320 FORMAT (// 47H ** WARNING.  ESTIMATE OF STORAGE FOR A DYNAMIC,
     1           32H ANALYSIS EXCEEDS AVAILABLE CORE, // 1X)
C
 1001 FORMAT (14I5)
      END
C&===================================================================
      SUBROUTINE ADDSTF (A,B,STR,TMASS,NUMEL,NBLOCK,NE2B,LL,MBAND,ANORM,
     1NVV)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  MAIN
C
C     FORMS GLOBAL EQUILIBRIUM EQUATIONS IN BLOCKS
C
      DIMENSION A(NE2B,MBAND),B(NE2B,LL),STR(4,LL),TMASS(NE2B)
C
      COMMON /DYN/   NT,NOT,ALFA,DT,BETA,NFN,NGM,NAT,NDYN
      COMMON /EM/ LRD,ND,LM(63),IPAD,SS(2331)
      COMMON /EXTRA/ MODEX,NT8,IFILL(14)
C
      NEQB=NE2B/2
      K=NEQB+1
      X=NBLOCK
      MB=DSQRT(X)
      MB=MB/2+1
      NEBB=MB*NE2B
      MM=1
      NDEG=0
      NVV=0
      ANORM=0.
      NSHIFT=0
      REWIND 3
      REWIND 4
      REWIND 9
C
C     READ ELEMENT LOAD MULTIPLIERS
C
      WRITE (33,2000)
      DO 50 L=1,LL
      READ  (5,1002)   (STR(I,L),I=1,4)
   50 WRITE (33,2002) L,(STR(I,L),I=1,4)
      IF(MODEX.EQ.0) WRITE (8) STR
C
C     FOR A STEP-BY-STEP ANALYSIS  (NDYN.EQ.4)  READ THE SOLUTION
C     CONTROL CARD.  THE TIME STEP (DT) AND THE DAMPING COEFFICIENTS
C     (ALFA/BETA) ARE REQUIRED FOR THE ASSEMBLY OF THE EFFECTIVE
C     SYSTEM STIFFNESS MATRIX IN THIS ROUTINE.
C
      IF(NDYN.NE.4) GO TO 65
C
      READ  (5,1004) NFN,NGM,NAT,NT,NOT,DT,ALFA,BETA
      WRITE (33,2004) NFN,NGM,NAT,NT,NOT,DT,ALFA,BETA
      IF(NAT.EQ.0) NAT = 1
      IF(NOT.EQ.0) NOT = 1
      IF(DT.GT.1.0E-12) GO TO 55
      WRITE (33,3000)
      STOP
C
C     COMPUTE INTEGRATION COEFFICIENTS FOR ASSEMBLY OF EFFECTIVE
C     SYSTEM STIFFNESS (STEP-BY-STEP ANALYSIS ONLY)
C
```

```
   55 TETA = 1.4
      DT1  = TETA*DT
      DT2  = DT1**2
      A0   = (6.+3.*ALFA*DT1)/(DT2+3.*BETA*DT1)
C
   65 IF(MODEX.EQ.1) RETURN
C
C     FORM EQUATIONS IN BLOCKS     ( 2 BLOCKS AT A TIME)
C
      DO 1000 M=1,NBLOCK ,2
      DO 100 I=1,NE2B
      DO 100 J=1,MBAND
  100 A(I,J)=0.
      READ (3) ((B(I,L),I=1,NEQB),L=1,LL),(TMASS(I),I=1,NEQB)
      IF (M.EQ.NBLOCK) GO TO 200
      READ (3) ((B(I,L),I=K,NE2B),L=1,LL),(TMASS(I),I=K,NE2B)
  200 CONTINUE
C
      REWIND 55
      REWIND 2
      NA=55
      NUME=NUM7
      IF (MM.NE.1) GO TO 75
      NA=2
      NUME=NUMEL
      NUM7 =0
C
   75 DO 700 N=1,NUME
      READ (NA) LRD,ND,(LM(I),I=1,ND),(SS(I),I=1,LRD)
      MSHFT = ND * (ND+1)/2 +4 *ND
      DO 600 I=1,ND
      LMN=1-LM(I)
      II=LM(I)-NSHIFT
      IF (II.LE.0.OR.II.GT.NE2B) GO TO 600
      IMS=I+MSHFT
      TMASS(II)=TMASS(II)+ SS(IMS)
      DO 300 L=1,LL
      DO 300 J=1,4
      KK = ND *(ND+1)/2 + ND*(J-1)
  300 B(II,L)=B(II,L)+SS(I+KK)*STR(J,L)
      DO 500 J=1,ND
      JJ=LM(J)+LMN
      IF(JJ) 500,500,390
  390 IF(J-I) 396,394,394
  394 KK = ND*I-(I-1)*I/2 +J-ND
      GO TO 400
  396 KK =ND*J -(J-1)*J/2+I-ND
  400 A(II,JJ)=A(II,JJ)+SS(  KK)
  500 CONTINUE
  600 CONTINUE
C
C     DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 55
C
      IF (MM.GT.1) GO TO 700
```

```
      DO 650 I=1,ND
      II=LM(I) -NSHIFT
      IF(II.GT.NE2B.AND.II.LE.NEBB) GO TO 660
  650 CONTINUE
      GO TO 700
  660 WRITE (55) LRD,ND,(LM(I),I=1,ND),(SS(I),I=1,LRD)
      NUM7=NUM7+1
C
  700 CONTINUE
      DO 710 L=1,NEQB
      ANORM=ANORM + A(L,1)
      IF (A(L,1).NE.0.) NDEG=NDEG + 1
      IF (A(L,1).EQ.0.) A(L,1)=1.E+20
      IF (TMASS(L).NE.0.) NVV=NVV + 1
  710 CONTINUE
C
C     FOR STEP-BY-STEP ANALYSIS ADD THE MASS CONTRIBUTIONS TO
C     THE EQUATION DIAGONAL COEFFICIENTS
C
      IF(NDYN.NE.4) GO TO 716
      DO 714 I=1,NEQB
  714 A(I,1) = A(I,1) + AO* TMASS(I)
      WRITE (4) ((A(I,J),I=1,NEQB),J=1,MBAND)
      GO TO 718
  716 WRITE (4) ((A(I,J),I=1,NEQB),J=1,MBAND),((B(I,L),I=1,NEQB),L=1,LL)
  718 WRITE (9)  (TMASS(I),I=1,NEQB)
C
      IF(M.EQ.NBLOCK) GO TO 1000
      DO 720 L=K,NE2B
      ANORM=ANORM + A(L,1)
      IF (A(L,1).NE.0.) NDEG=NDEG + 1
      IF (A(L,1).EQ.0.) A(L,1)=1.E+20
      IF (TMASS(L).NE.0.) NVV=NVV + 1
  720 CONTINUE
C
      IF(NDYN.NE.4) GO TO 726
      DO 724 I=K,NE2B
  724 A(I,1) = A(I,1) + AO* TMASS(I)
      WRITE (4) ((A(I,J),I=K,NE2B),J=1,MBAND)
      GO TO 728
  726 WRITE (4) ((A(I,J),I=K,NE2B),J=1,MBAND),((B(I,L),I=K,NE2B),L=1,LL)
  728 WRITE (9)  (TMASS(I),I=K,NE2B)
C
      IF (MM.EQ.MB) MM=0
      MM=MM+1
 1000 NSHIFT=NSHIFT+NE2B
      IF (NDEG.GT.0) GO TO 730
      WRITE (33,1010)
      STOP
  730 ANORM=(ANORM/NDEG)*1.E-8
C
      RETURN
 1002 FORMAT (4F10.0)
 1004 FORMAT (5I5,3F10.0)
```

```
 1010 FORMAT (51HOSTRUCTURE WITH NO DEGREES OF FREEDOM  CHECK DATA   )
 2000 FORMAT (/// 10H STRUCTURE,13X,7HELEMENT,4X,4HLOAD,4X,
     1 11HMULTIPLIERS,/ 10H LOAD CASE,12X,1HA,9X,1HB,9X,1HC,9X,1HD,/ 1X)
 2002 FORMAT (I6,7X,4F10.3)
 2004 FORMAT (45H1S T E P - B Y - S T E P   S O L U T I O N   ,
     1         37HC O N T R O L   I N F O R M A T I O N, ///
     2 5X, 35HNUMBER OF TIME VARYING FUNCTIONS  =, I5    //
     3 5X, 35HGROUND MOTION INDICATOR           =, I5    /
     4 8X, 10HEQ.0, NONE, /
     5 8X, 29HGT.0, READ ACCELERATION INPUT, //
     6 5X, 35HNUMBER OF ARRIVAL TIMES           =, I5    /
     7 8X, 26HEQ.0, ALL FUNCTIONS ARRIVE, /
     8 8X, 18H      AT TIME ZERO, //
     9 5X, 35HNUMBER OF SOLUTION TIME STEPS     =, I5    //
     A 5X, 35HOUTPUT (PRINT) INTERVAL           =, I5    //
     B 5X, 35HSOLUTION TIME INCREMENT           =, E14.4 //
     C 5X, 30HMASS-     PROPORTIONAL DAMPING, /
     D 5X, 35HCOEFFICIENT (ALPHA)               =, E14.4 //
     E 5X, 30HSTIFFNESS-PROPORTIONAL DAMPING, /
     F 5X, 35HCOEFFICIENT (BETA)                =, E14.4 /// 1X)
 3000 FORMAT (27H0*** ERROR   ZERO TIME STEP, / 1X)
      END
C
C
C&=======================================================================
      SUBROUTINE BOUND
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON A(1)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /JUNK/  LT,LH,L,IPAD,SIG(20),IFILL(386)
      COMMON /EXTRA/ MODEX,NT8,N10SV,NT10,IFILL2(12)
      IF (NPAR(1).EQ.0) GO TO 500
      CALL CLAMP (NPAR(2),A(N1),A(N2),A(N3),A(N4),NUMNP,MBAND)
      RETURN
  500 continue
c--  WRITE (33,2002)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
c--   WRITE (33,2001)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
c--   WRITE (33,3002) MM,L,(SIG(I),I=1,2)   !printing suppressed
      IF(N10SV.EQ.1)
     *WRITE (NT10) MM,L,SIG(1),SIG(2)
  800 CONTINUE
      RETURN
 2001 FORMAT (/)
 2002 FORMAT (48H1B O U N D A R Y   E L E M E N T   F O R C E S /,
     1         14H M O M E N T S, // 8H ELEMENT,3X,4HLOAD,14X,5HFORCE,
     2         9X,6HMOMENT, / 8H  NUMBER,3X,4HCASE,  // 1X)
 3002 FORMAT  (I8,I7,4X,2E15.5)
      END
C
```

```
C
C&=====================================================================
      SUBROUTINE CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM,NS)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  RUSS,TEAM,PLNAX,BRICK8,TPLATE,CLAMP,ELST3D,PIPEK
C
C-----CALCULATES BAND WIDTH AND WRITES STIFFNESS MATRIX ON TAPE 2
      DIMENSION LM(1),XM(1),S(NDM,NDM),P(NDM,4)
      COMMON /EXTRA/ MODEX,NT8,IFILL(14)
      MIN=100000
      MAX=0
      DO 800 L=1,ND
      IF (LM(L).EQ.0) GO TO 800
      IF (LM(L).GT.MAX) MAX=LM(L)
      IF (LM(L).LT.MIN) MIN=LM(L)
  800 CONTINUE
      NDIF=MAX-MIN+1
      IF (NDIF.GT.MBAND) MBAND=NDIF
      IF(MODEX.EQ.1) GO TO 810
C
      LRD=ND*(ND+1)/2+5*ND
      WRITE(2) LRD,ND,(LM(I),I=1,ND),((S(I,J),J=I,ND),I=1,ND),
     1 ((P(I,J),I=1,ND),J=1,4),(XM(I),I=1,ND)
      RETURN
C
  810 WRITE (1) ND,NS,(LM(I),I=1,ND)
      RETURN
C
      END
C
C
C&======================================================================
      SUBROUTINE CLAMP (NUMEL,ID,X,Y,Z,NUMNP,MBAND)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),SA(12,24),TT(12,4),
     1    IFILL1(3048)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
      COMMON / JUNK / R(6),RM(4),IFILL2(410)
      COMMON /EXTRA/ MODEX,NT8,IFILL3(14)
      WRITE (33,2000) NUMEL
      NS=2
      ND=6
      READ(5,1005) RM
      WRITE (33,2005) RM
      IF(MODEX.EQ.1)
     *WRITE (NT8) RM
      DO 30 NI=1,ND
      XM(NI) = 0.0
      DO 20 NJ=1,ND
   20 S(NI,NJ)=0.0
   30 CONTINUE
      DO 50 NK=1,NS
      DO 40 NL=1,ND
```

```
 1010 FORMAT (51HOSTRUCTURE WITH NO DEGREES OF FREEDOM  CHECK DATA    )
 2000 FORMAT (/// 10H STRUCTURE,13X,7HELEMENT,4X,4HLOAD,4X,
     1 11HMULTIPLIERS,/ 10H LOAD CASE,12X,1HA,9X,1HB,9X,1HC,9X,1HD,/ 1X)
 2002 FORMAT (I6,7X,4F10.3)
 2004 FORMAT (45H1S T E P - B Y - S T E P   S O L U T I O N    ,
     1          37HC O N T R O L   I N F O R M A T I O N, ///
     2 5X, 35HNUMBER OF TIME VARYING FUNCTIONS  =, I5    //
     3 5X, 35HGROUND MOTION INDICATOR           =, I5    /
```

```
   40 SA(NK,NL) = 0.0
      DO 50 NI=1,4
      TT(NK,NI) =0.0
   50 CONTINUE
      NE=0
      WRITE (33,2007)
  210 KG=0
      MARK=0
  200 READ (5,1000) NP,NI,NJ,NK,NL,KD,KR,KN,SD,SR,TRACE
      IF (TRACE.EQ.0.) TRACE=1.0E+10
      IF (KG.GT.0) GO TO 550
      KG=KN
      IF(MODEX.EQ.1) GO TO 530
      IF(NJ.EQ.0)GO TO 250
      X1=X(NJ)-X(NI)
      Y1=Y(NJ)-Y(NI)
      Z1=Z(NJ)-Z(NI)
      X2=X(NL)-X(NK)
      Y2=Y(NL)-Y(NK)
      Z2=Z(NL)-Z(NK)
      T1=Y1*Z2-Y2*Z1
      T2=Z1*X2-Z2*X1
      T3=X1*Y2-X2*Y1
      GO TO 260
  250 T1=X(NI)-X(NP)
      T2=Y(NI)-Y(NP)
      T3=Z(NI)-Z(NP)
  260 XL=T1*T1+T2*T2+T3*T3
      XL=DSQRT(XL)
      IF(XL.GT.1.0E-6) GO TO 270
      WRITE (33,3000)
 3000 FORMAT (32HO*** ERROR   ZERO ELEMENT LENGTH, / 1X)
      STOP
  270 CONTINUE
      T1=T1/XL
      T2=T2/XL
      T3=T3/XL
      IF (KD.EQ.0) GO TO 300
      SA(1,1)=T1*TRACE
      SA(1,2)=T2*TRACE
      SA(1,3)=T3*TRACE
      S(1,1)=T1*T1*TRACE
      S(1,2)=T1*T2*TRACE
      S(1,3)=T1*T3*TRACE
      S(2,2)=T2*T2*TRACE
      S(2,3)=T2*T3*TRACE
      S(3,3)=T3*T3*TRACE
      PP=TRACE*SD
      R(1)=T1*PP
      R(2)=T2*PP
      R(3)=T3*PP
      GO TO 350
  300 DO 310 I=1,3
      R(I)=0.0
```

```
      SA(1,I)=0.0
      DO 310 J=1,3
  310 S(I,J)=0.0
  350 IF (KR.EQ.0) GO TO 400
      SA(2,5)=T2*TRACE
      SA(2,4)=T1*TRACE
      SA(2,6)=T3*TRACE
      S(4,4)=T1*T1*TRACE
      S(4,5)=T1*T2*TRACE
      S(4,6)=T1*T3*TRACE
      S(5,5)=T2*T2*TRACE
      S(5,6)=T2*T3*TRACE
      S(6,6)=T3*T3*TRACE
      PP=TRACE*SR
      R(4)=T1*PP
      R(5)=T2*PP
      R(6)=T3*PP
      GO TO 450
  400 DO 410 I=4,6
      R(I)=0.0
      SA(2,I)=0.0
      DO 410 J=I,6
  410 S(I,J)=0.0
  450 DO 500 I=1,ND
      DO 500 J=1,ND
  500 S(J,I)=S(I,J)
      DO 520 I=1,ND
      DO 520 J=1,4
  520 P(I,J)=R(I)*RM(J)
  530 NN=NP
      NNI=NI
      NNJ=NJ
      NNK=NK
      NNL=NL
      NKD=KD
      NKR=KR
      SSD=SD
      SSR=SR
      TTR=TRACE
      GO TO 560
  550 MARK=1
  555 NN=NN+KG
      NNI=NNI+KG
  560 KEL=NE+1
      WRITE (33,2010) KEL,NN,NNI,NNJ,NNK,NNL,NKD,NKR,KN,SSD,SSR,TTR
      NE=NE+1
      IF(MODEX.EQ.1)
     *WRITE(NT8) NE,NN,NNI,NNJ,NNK,NNL,NKD,NKR,SSD,SSR,TTR
      DO 600 I=1,ND
  600 LM(I)=ID(NN,I)
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM,NS)
      IF(MODEX.EQ.1) GO TO 650
      WRITE (1) ND,NS,(LM(L),L=1,ND),((SA(L,K),L=1,NS),K=1,ND),
```

```
     1 ((TT(L,K),L=1,NS),K=1,4)
 650 CONTINUE
      IF (NE.EQ.NUMEL) RETURN
      IF (NN.LT.NP) GO TO 555
      IF (MARK.EQ.1) GO TO 210
      GO TO 200
1000 FORMAT (8I5,3F10.0)
1005 FORMAT (4F10.0)
2000 FORMAT (34H1B O U N D A R Y   E L E M E N T S, ///
     1          27H ELEMENT TYPE        =      7, /
     2           21H NUMBER OF ELEMENTS =,I6     ///1X)
2005 FORMAT (30H ELEMENT LOAD CASE MULTIPLIERS, // 8X,7HCASE(A),8X,
     1          7HCASE(B),8X,7HCASE(C),8X,7HCASE(D),/ 4E15.4 /// 1X)
2007 FORMAT (53H ELEMENT   NODE   NODES DEFINING CONSTRAINT DIRECTION,
     1      3X,38HCODE   CODE  GENERATION       SPECIFIED,6X,
     2          22HSPECIFIED        SPRING, /
     3          53H  NUMBER    (N)        (NI)      (NJ)      (NK)      (NL),
     4      3X,38H  KD      KR   CODE (KN)   DISPLACEMENT,6X,
     5          22H ROTATION            RATE, / 1X)
2010 FORMAT (1X,2(2X,I5),2X,4(4X,I5),2(2X,I5),7X,I5,2E15.4,E13.4)
      END
C
C&======================================================================
      SUBROUTINE CROSS2 (A,B,C,IERR)
C
C     CALLED BY : INP21
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     THIS ROUTINE FORMS THE VECTOR PRODUCT   C = A*B   WHERE *C*
C     IS NORMALIZED TO UNIT LENGTH
C
      DIMENSION A(3),B(3),C(3)
C
      X = A(2) * B(3) - A(3) * B(2)
      Y = A(3) * B(1) - A(1) * B(3)
      Z = A(1) * B(2) - A(2) * B(1)
      XLN =DSQRT(X*X+Y*Y+Z*Z)
      IERR = 1
      IF(XLN.LE.1.0E-08) RETURN
      XLN = 1.0 /XLN
      C(3) = Z * XLN
      C(2) = Y * XLN
      C(1) = X * XLN
      IERR = 0
      RETURN
      END
C&======================================================================
      SUBROUTINE DER3DS (NEL,XX,B,DET,R,S,T,NOD9,H,P,IELD,IELX)
C
C     CALLED BY : THDEE
C
      IMPLICIT REAL*8(A-H,O-Z)
C
```

```
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .   P R O G R A M                                                    .
C .                                                                    .
C .      EVALUATES STRAIN-DISPLACEMENT MATRIX B AT POINT (R,S,T)       .
C .                                                                    .
C .      CURVILINEAR HEXAHEDRON   8 TO 21 NODES                        .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C
      DIMENSION  XX(3,1),B(6,1),NOD9(1),H(1),P(3,1)
      DIMENSION  XJ(3,3),XJI(3,3)
C
C
C     FIND INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
C     EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)
C     COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)
C
C
      CALL FNCT (R,S,T,H,P,NOD9,XJ,DET,XX,IELD,IELX,NEL)
C
C
C     COMPUTE INVERSE OF JACOBIAN MATRIX
C
C
      DUM=1.0/DET
      XJI(1,1)=DUM*( XJ(2,2)*XJ(3,3) - XJ(2,3)*XJ(3,2))
      XJI(2,1)=DUM*(-XJ(2,1)*XJ(3,3) + XJ(2,3)*XJ(3,1))
      XJI(3,1)=DUM*( XJ(2,1)*XJ(3,2) - XJ(2,2)*XJ(3,1))
      XJI(1,2)=DUM*(-XJ(1,2)*XJ(3,3) + XJ(1,3)*XJ(3,2))
      XJI(2,2)=DUM*( XJ(1,1)*XJ(3,3) - XJ(1,3)*XJ(3,1))
      XJI(3,2)=DUM*(-XJ(1,1)*XJ(3,2) + XJ(1,2)*XJ(3,1))
      XJI(1,3)=DUM*( XJ(1,2)*XJ(2,3)  - XJ(1,3)*XJ(2,2))
      XJI(2,3)=DUM*(-XJ(1,1)*XJ(2,3) + XJ(1,3)*XJ(2,1))
      XJI(3,3)=DUM*( XJ(1,1)*XJ(2,2) - XJ(1,2)*XJ(2,1))
C
C
C     EVALUATE B MATRIX IN GLOBAL (X,Y,Z) COORDINATES
C
C
      DO 130 K=1,IELD
      K2=K*3
      DO 115 L=1,3
      B(L,K2-2) = 0.0
      B(L,K2-1) = 0.0
  115 B(L,K2  ) = 0.0
C
C     DIRECT STRAINS  (1=EXX, 2=EYY, 3=EZZ)
C
      DO 120 I=1,3
```

```
      B(1,K2-2) = B(1,K2-2) + XJI(1,I)* P(I,K)
      B(2,K2-1) = B(2,K2-1) + XJI(2,I)* P(I,K)
  120 B(3,K2  ) = B(3,K2  ) + XJI(3,I)* P(I,K)
C
C     SHEAR STRAINS  (4=EXY, 5=EYZ, 6=EZX)
C
      B(4,K2-2) = B(2,K2-1)
      B(4,K2-1) = B(1,K2-2)
      B(5,K2-1) = B(3,K2  )
      B(5,K2  ) = B(2,K2-1)
      B(6,K2-2) = B(3,K2  )
  130 B(6,K2  ) = B(1,K2-2)
C
C
      RETURN
C
      END
C&=======================================================================
      SUBROUTINE ELTYPE(MTYPE)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  MAIN,STRESS
C
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12),MTYPE
C
C     THREE DIMENSIONAL TRUSS ELEMENTS
C
    1 CONTINUE
C   1 CALL TRUSS
      GO TO 900
C
C     THREE DIMENSIONAL BEAM ELEMENTS
C
    2 CONTINUE
C   2 CALL BEAM
      GO TO 900
C
C     PLANE STRESS ELEMENTS
C
    3 CONTINUE
C   3 CALL PLANE
      GO TO 900
C
C     AXISYMMETRIC SOLID ELEMENTS
C
    4 CONTINUE
C   4 CALL PLANE
      GO TO 900
C
C     THREE DIMENSIONAL SOLID ELEMENTS
C
    5 CONTINUE
C   5 CALL THREED
```

```
      GO TO 900
C
C     PLATE BENDING ELEMENTS
C
    6 CONTINUE
C   6 CALL SHELL
      GO TO 900
C
C
    7 CALL BOUND
      GO TO 900
C
C     THICK SHELL ELEMENTS
C
    8 CALL SOL21
      GO TO 900
C
    9 WRITE (33,100) MTYPE
      GO TO 900
C
   10 WRITE (33,100) MTYPE
      GO TO 900
C
   11 WRITE (33,100) MTYPE
      GO TO 900
C
C     STRAIGHT OR CURVED PIPE ELEMENTS
C
   12 CONTINUE
C  12 CALL PIPE
C
  900 RETURN
C
  100 FORMAT ('0ELEMENT',I4,' IS NOT IMPLEMENTED YET')
      END
C&===================================================================
      SUBROUTINE ERROR(N)
      WRITE (33,2000) N
 2000 FORMAT (// 20H STORAGE EXCEEDED BY  I6)
      STOP
      END
C&===================================================================
      SUBROUTINE FACEPR (NEL,KDIS,KXYZ,XX,NOD9,H,P,PL,NFACE,LT,PWA,KLS)
C
C     CALLED BY : THDFE
C     CALLS : FNCT
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C     THIS ROUTINE COMPUTES NODE FORCES DUE TO APPLIED ELEMENT FACE
C     PRESSURE DISTRIBUTIONS
C
C
```

```
      DIMENSION    XX(3,1),NOD9(1),H(1),P(3,1),PL(1),PWA(1)
      DIMENSION    XJ(3,3),ETA(3),KFACE(6,8),KCRD(6),FVAL(6),IPRM(3),
     1             PR(8),NODES(8),IPR4(4)
      COMMON /GAUSS/ XK(4,4),WGT(4,4)
C
      DATA KFACE / 1, 2, 1, 4, 1, 5,
     1              4, 3, 5, 8, 2, 6,
     2              8, 7, 6, 7, 3, 7,
     3              5, 6, 2, 3, 4, 8,
     4             12, 10, 17, 20,  9, 13,
     5             20, 19, 13, 15, 10, 14,
     6             16, 14, 18, 19, 11, 15,
     7             17, 18,  9, 11, 12, 16/
C
      DATA  KCRD  /  1,  1,  2,  2,  3,  3/
      DATA  FVAL  / 1.,-1., 1.,-1., 1.,-1./
      DATA  IPRM  /  2,  3,  1/
      DATA IPR4  /  2,  3,  4,  1/
C
C     DETERMINE THE ELEMENT NODES CONTRIBUTING TO FORCE CALCULATIONS
C     ON THIS FACE
C
      DO 2 I=1,4
      NODES(I ) = KFACE(NFACE,I)
      NODES(I+4) = 0
    2 CONTINUE
C
      IF(KDIS.LT.9) GO TO 9
C
      NN9 = KDIS-8
C
      DO 8 K=5,8
      DO 4 I=1,NN9
C
      J = I
      IF(KFACE(NFACE,K).EQ.NOD9(I)) GO TO 6
C
    4 CONTINUE
      GO TO 8
C
    6 NODES(K) = J
    8 CONTINUE
C
    9 CONTINUE
C
C     SET UP THE PRESSURE VECTOR FOR THE FOUR FACE CORNER NODES
C
C        1. ADJUST THE SIGN OF THE PRESSURES SO THAT POSITIVE
C           PRESSURE ALWAYS COMPRESSES THE ELEMENT
C
      FACT = -FVAL(NFACE)
C
      GO TO (10,30), LT
C
```

```
C           2. DISTRIBUTED PRESSURE GIVEN AT THE CORNER NODES
C
   10 DO 25 K=1,8
C
      IF(NODES(K).EQ.0) GO TO 25
C
      IF(K.GT.4) GO TO 15
C
      PR(K) = PWA(K) * FACT
      GO TO 25
C
   15 J = K-4
      L = IPR4(J)
      PR(K) = (PWA(J) + PWA(L)) * 0.5 * FACT
C
   25 CONTINUE
      GO TO 75
C
C           3. ELEMENT FACE EXPOSED TO HYDROSTATIC PRESSURE
C
   30 GAMMA = PWA(1)* FACT
C
      XLN = 0.0
      DO 35 K=1,3
      ETA(K) =  PWA(K+4) - PWA(K+1)
   35 XLN = XLN + ETA(K)**2
      XLN =DSQRT(XLN)
C
      IF(XLN.GT.1.0E-6) GO TO 40
C
      WRITE (33,3000) KLS,NEL
 3000 FORMAT (31H0ERROR***   PRESSURE LOAD SET (,I3,15H) FOR ELEMENT (,
     1          I5,43H) HAS UNDEFINED HYDROSTATIC SURFACE NORMAL., / 1X)
      STOP
C
   40 DO 45 K=1,3
   45 ETA(K) = ETA(K)/ XLN
C
      DO 70 N=1,8
C
      IF(NODES(N).EQ.0) GO TO 70
C
      XLN = 0.0
                NOD = NODES(N)
      IF(N.GT.4) NOD = NOD + 8
C
      DO 50 I=1,3
   50 XLN = XLN + (XX(I,NOD) - PWA(I+1))* ETA(I)
C
      PR(N) = XLN* GAMMA
C
      IF(XLN.LT.0.0) PR(N) = 0.0
C
   70 CONTINUE
```

```
   75 CONTINUE
C
C     SET UP VARIABLES FOR THE SURFACE INTEGRATION
C
      ML = KCRD(NFACE)
      MM = IPRM(ML)
      MN = IPRM(MM)
C
C     SURFACE INTEGRATION LOOP
C
      ETA(ML) = EVAL(NFACE)
C
      DO 300 LX=1,3
C
      ETA(MM) = XK(LX,3)
C
      DO 300 LY=1,3
C
      ETA(MN) = XK(LY,3)
C
      WT = WGT(LX,3)* WGT(LY,3)
C
C     EVALUATE THE INTERPOLATION FUNCTIONS AND JACOBIAN MATRIX
C
      CALL FNCT (ETA(1),ETA(2),ETA(3),H,P,NOD9,XJ,DET,XX,KDIS,KK12,NEL)
C
C     COMPUTE THE DIRECTION COSINES OF THE UNIT SURFACE NORMAL VECTOR
C     AT THIS SAMPLE POINT
C
      A1 = XJ(MM,2)* XJ(MN,3)- XJ(MM,3)* XJ(MN,2)
      A2 = XJ(MM,3)* XJ(MN,1)- XJ(MM,1)* XJ(MN,3)
      A3 = XJ(MM,1)* XJ(MN,2)- XJ(MM,2)* XJ(MN,1)
C
      AA =DSQRT(A1**2 + A2**2 + A3**2)
      IF(AA.GT.1.0E-8) GO TO 100
C
      WRITE (33,3010) NFACE,NEL
 3010 FORMAT (38H0ERROR***   UNDEFINED NORMAL IN FACE (,I1,5H) FOR,
     1          10H ELEMENT (,I5,2H)., / 1X)
      STOP
C
  100 FACT = 1.0/AA
      A1 = A1* FACT
      A2 = A2* FACT
      A3 = A3* FACT
C
C     COMPUTE THE FIRST FUNDAMENTAL FORM (AREA DIFFERENTIAL)
C
      AA = 0.0
      BB = 0.0
      CC = 0.0
      DO 120 I=1,3
      AA = AA + XJ(MM,I)**2
      CC = CC + XJ(MN,I)**2
```

```
  120 BB = BB + XJ(MM,I)* XJ(MN,I)
      C  =DSQRT(AA*CC - BB**2)
C
C     INTERPOLATE FOR THE PRESSURE AT THIS SAMPLE POINT
C
      PRESS = 0.0
C
      DO 130 K=1,8
C
      IF(NODES(K).EQ.0) GO TO 130
C
               NOD = NODES(K)
      IF(K.GT.4) NOD = NOD + 8
C
      PRESS = PRESS + H(NOD)* PR(K)
  130 CONTINUE
C
      FACT = WT* C* PRESS
C
C     ASSEMBLE THE NODE FORCE CONTRIBUTION
C
      DO 160 L=1,8
C
      IF(NODES(L).EQ.0) GO TO 160
C
      IF(L.GT.4) GO TO 140
C
C        1. CORNER NODES
C
      N = NODES(L)
      K = 3*N
      GO TO 150
C
C        2. SIDE NODES
C
  140 J = NODES(L)
      N = J+8
      K = 3* NOD9(J)
C
  150 QQ = FACT* H(N)
C
      PL(K-2) = PL(K-2) + QQ* A1
      PL(K-1) = PL(K-1) + QQ* A2
      PL(K  ) = PL(K  ) + QQ* A3
  160 CONTINUE
C
  300 CONTINUE
C
      RETURN
      END
C&=====================================================================
      SUBROUTINE FNCT (R,S,T,H,P,NOD9,XJ,DET,XX,IELD,IELX,NEL)
C
C     CALLED BY : FACEPR
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                .
C .   P R O G R A M                                                .
C .                                                                .
C .      TO FIND INTERPOLATION FUNCTIONS ( H )                     .
C .      AND DERIVATIVES ( P ) CORRESPONDING TO THE NODAL          .
C .      POINTS OF A CURVILINEAR ISOPARAMETRIC HEXAHEDRON          .
C .      OR SUBPARAMETRIC HEXAHEDRON (8 TO 21 NODES)               .
C .                                                                .
C .      TO FIND JACOBIAN ( XJ ) AND ITS DETERMINANT ( DET )       .
C .                                                                .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
      DIMENSION H(1),P(3,1),NOD9(1),IPERM(8),XJ(3,3),XX(3,1)
C
      DATA IPERM / 2,3,4,1,6,7,8,5 /
C
      IEL = IELD
      NND9= IELD-8
C
      RP=1.0 + R
      SP=1.0 + S
      TP=1.0 + T
      RM=1.0 - R
      SM=1.0 - S
      TM=1.0 - T
      RR=1.0 - R*R
      SS=1.0 - S*S
      TT=1.0 - T*T
C
C
C     INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
C
C
C     8-NODE BRICK
C
      H(1)=0.125*RP*SP*TP
      H(2) =0.125*RM*SP*TP
      H(3)=0.125*RM*SM*TP
      H(4)=0.125*RP*SM*TP
      H(5)=0.125*RP*SP*TM
      H(6)=0.125*RM*SP*TM
      H(7)=0.125*RM*SM*TM
      H(8)=0.125*RP*SM*TM
C
      P(1,1)=0.125*SP*TP
      P(1,2)=-P(1,1)
      P(1,3)=-0.125*SM*TP
      P(1,4)=-P(1,3)
```

```
      P(1,5)=0.125*SP*TM
      P(1,6)=-P(1,5)
      P(1,7)=-0.125*SM*TM
      P(1,8)=-P(1,7)
C
      P(2,1)=0.125*RP*TP
      P(2,2)=0.125*RM*TP
      P(2,3)=-P(2,2)
      P(2,4)=-P(2,1)
      P(2,5)=0.125*RP*TM
      P(2,6)=0.125*RM*TM
      P(2,7)=-P(2,6)
      P(2,8)=-P(2,5)
C
      P(3,1)=0.125*RP*SP
      P(3,2)=0.125*RM*SP
      P(3,3)=0.125*RM*SM
      P(3,4)=0.125*RP*SM
      P(3,5)=-P(3,1)
      P(3,6)=-P(3,2)
      P(3,7)=-P(3,3)
      P(3,8)=-P(3,4)
C
      IF(IEL.EQ.8) GO TO 50
C
C
C     ADD DEGREES OF FREEDOM IN EXCESS OF 8
C
      I=0
    2 I=I + 1
      IF(I.GT.NND9) GO TO 40
      NN=NOD9(I) - 8
      GO TO (9,10,11,12,13,14,15,16,17,18,19,20,21) ,NN
C
    9 H(9) = 0.25*RR*SP*TP
      P(1,9) = -0.50*R*SP*TP
      P(2,9) = 0.25*RR*TP
      P(3,9) = 0.25*RR*SP
      GO TO 2
   10 H(10)=0.25*RM*SS*TP
      P(1,10)=-0.25*SS*TP
      P(2,10)=-0.50*RM*S*TP
      P(3,10)= 0.25*RM*SS
      GO TO 2
   11 H(11)=0.25*RR*SM*TP
      P(1,11)=-0.50*R*SM*TP
      P(2,11)=-0.25*RR*TP
      P(3,11)= 0.25*RR*SM
      GO TO 2
   12 H(12)=0.25*RP*SS*TP
      P(1,12)= 0.25*SS*TP
      P(2,12)=-0.50*RP*S*TP
      P(3,12)= 0.25*RP*SS
      GO TO 2
```

```
   13 H(13)=0.25*RR*SP*TM
      P(1,13)=-0.50*R*SP*TM
      P(2,13)= 0.25*RR*TM
      P(3,13)=-0.25*RR*SP
      GO TO 2
   14 H(14)=0.25*RM*SS*TM
      P(1,14)=-0.25*SS*TM
      P(2,14)=-0.50*RM*S*TM
      P(3,14)=-0.25*RM*SS
      GO TO 2
   15 H(15)=0.25*RR*SM*TM
      P(1,15)=-0.50*R*SM*TM
      P(2,15)=-0.25*RR*TM
      P(3,15)=-0.25*RR*SM
      GO TO 2
   16 H(16)=0.25*RP*SS*TM
      P(1,16)= 0.25*SS*TM
      P(2,16)=-0.50*RP*S*TM
      P(3,16)=-0.25*RP*SS
      GO TO 2
   17 H(17)=0.25*RP*SP*TT
      P(1,17)=0.25*SP*TT
      P(2,17)=0.25*RP*TT
      P(3,17)=-0.50*RP*SP*T
      GO TO 2
   18 H(18)=0.25*RM*SP*TT
      P(1,18)=-0.25*SP*TT
      P(2,18)= 0.25*RM*TT
      P(3,18)=-0.50*RM*SP*T
      GO TO 2
   19 H(19)=0.25*RM*SM*TT
      P(1,19)=-0.25*SM*TT
      P(2,19)=-0.25*RM*TT
      P(3,19)=-0.50*RM*SM*T
      GO TO 2
   20 H(20)=0.25*RP*SM*TT
      P(1,20)= 0.25*SM*TT
      P(2,20)=-0.25*RP*TT
      P(3,20)=-0.50*RP*SM*T
      GO TO 2
   21 H(21)=RR*SS*TT
      P(1,21)=-2.0*R*SS*TT
      P(2,21)=-2.0*S*RR*TT
      P(3,21)=-2.0*T*RR*SS
      GO TO 2
C
C     MODIFY FIRST 8 FUNCTIONS IF 9 OR MORE NODES IN ELEMENT
C
   40 IH=0
   41 IH=IH + 1
      IF (IH.GT.NND9) GO TO 50
      II=IH + 7
      IF (II.EQ.IELX) GO TO 51
   42 IN=NOD9(IH)
```

```
      IF (IN.GT.16) GO TO 46
      I1=IN -8
      I2=IPERM(I1)
      H(I1)=H(I1) - 0.5*H(IN)
      H(I2)=H(I2) - 0.5*H(IN)
      H(IH+8)=H(IN)
      DO 45 J=1,3
      P(J,I1)=P(J,I1) - 0.5*P(J,IN)
      P(J,I2)=P(J,I2) - 0.5*P(J,IN)
   45 P(J,IH+8)=P(J,IN)
      GO TO 41
   46 IF (IN.EQ.21) GO TO 30
      I1=IN -16
      I2=I1 + 4
      H(I1)=H(I1) - 0.5*H(IN)
      H(I2)=H(I2) - 0.5*H(IN)
      H(IH+8)=H(IN)
      DO 47 J=1,3
      P(J,I1)=P(J,I1) - 0.5*P(J,IN)
      P(J,I2)=P(J,I2) - 0.5*P(J,IN)
   47 P(J,IH+8)=P(J,IN)
      GO TO 41
C
C     MODIFY FIRST 20 FUNCTIONS IF NODE 21 IS PRESENT
C
   30 IH=0
   31 IH=IH + 1
      IN=NOD9(IH)
      IF (IN.EQ.21) GO TO 35
      IF (IN.GT.16) GO TO 33
      I1=IN -8
      I2=IPERM(I1)
      H(I1)=H(I1) + 0.125*H(21)
      H(I2)=H(I2) + 0.125*H(21)
      DO 32 J=1,3
      P(J,I1)=P(J,I1) + 0.125*P(J,21)
   32 P(J,I2)=P(J,I2) + 0.125*P(J,21)
      GO TO 31
   33 I1=IN - 16
      I2=I1 + 4
      H(I1)=H(I1) + 0.125*H(21)
      H(I2)=H(I2) + 0.125*H(21)
      DO 34 J=1,3
      P(J,I1)=P(J,I1) + 0.125*P(J,21)
   34 P(J,I2)=P(J,I2) + 0.125*P(J,21)
      GO TO 31
   35 DO 36 I=1,8
      H(I)=H(I) - 0.125*H(21)
      DO 36 J=1,3
   36 P(J,I)=P(J,I) - 0.125*P(J,21)
      NN=NND9 + 7
      IF (NN.EQ.8) GO TO 50
      DO 38 I=9,NN
      H(I)=H(I) - 0.25*H(21)
```

```
      DO 38 J=1,3
   38 P(J,I)=P(J,I) - 0.25*P(J,21)
      H(NND9+8)=H(21)
      DO 39 J =1,3
   39 P(J,NND9+8)=P(J,21)
C
C
C     EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)
C
C
   50 IF (IELX.LT.IELD) RETURN
   51 DO 100 I=1,3
      DO 100 J=1,3
      DUM=0.0
      DO 90 K=1,IELX
   90 DUM=DUM + P(I,K)*XX(J,K)
  100 XJ(I,J)=DUM
C
C
C     COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)
C
C
      DET = XJ(1,1)*XJ(2,2)*XJ(3,3)
     1    + XJ(1,2)*XJ(2,3)*XJ(3,1)
     2    + XJ(1,3)*XJ(2,1)*XJ(3,2)
     3    - XJ(1,3)*XJ(2,2)*XJ(3,1)
     4    - XJ(1,2)*XJ(2,1)*XJ(3,3)
     5    - XJ(1,1)*XJ(2,3)*XJ(3,2)
      IF(DET.GT.1.0E-8) GO TO 110
      WRITE (33,2000) NEL,R,S,T
      STOP
  110 IF (IELX.LT.IELD) GO TO 42
C
C
      RETURN
C
C
C
 2000 FORMAT (49HOERROR***   NEGATIVE OR ZERO JACOBIAN DETERMINANT,
     1       23H COMPUTED FOR ELEMENT (,I5,1H), /
     2       12X,  3HR =, F10.5 /
     3     12X,  3HS =, F10.5 /
     4       12X,  3HT =, F10.5 / 1X)
C
C
      END
C&=======================================================================
      SUBROUTINE INL(ID,B,TR,TMASS,NUMNP,NEQB,LL)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  MAIN
C
C     INPUT NODAL LOADS AND MASSES
```

```
C
      DIMENSION ID(NUMNP,6),B(NEQB,LL),TR(6,LL),TMASS(NEQB)
      COMMON / JUNK / R(6),TXM(6),IFILL1(406)
      COMMON /EXTRA/ MODEX,NT8,IFILL2(14)
C
      NT=3
      REWIND NT
      KSHF=0
      WRITE (33,2002)
      IF(MODEX.EQ.1) GO TO 50
      DO 750 I=1,NEQB
      TMASS(I)=0.
      DO 750 K=1,LL
  750 B(I,K)=0.0
C
   50 DO 900 NN=1,NUMNP
C
      DO 100 I=1,6
      TXM(I)=0.
      DO 100 J=1,LL
  100 TR(I,J)=0.0
C
      IF(NN.EQ.1) GO TO 300
  150 IF(N.NE.NN) GO TO 400
      DO 200 I=1,6
      IF (L) 180,180,190
  180 TXM(I)=R(I)
      GO TO 200
  190 TR(I,L)=R(I)
  200 CONTINUE
  300 READ (5,1001) N,L,R
      IF (N.EQ.0) GO TO 150
      WRITE(33,2001) N,L,R
      GO TO 150
C
  400 IF(MODEX.EQ.1) GO TO 900
      DO 800 J=1,6
      II=ID(NN,J)-KSHF
      IF (II) 800,800,500
  500 DO 600  K=1,LL
  600 B(II,K)=TR(J,K)
      TMASS(II)=TXM(J)
  610 IF(II.NE.NEQB) GO TO 800
      WRITE (NT) B,TMASS
      KSHF=KSHF+NEQB
      DO 700 I=1,NEQB
      TMASS(I)=0.
      DO 700 K=1,LL
  700 B(I,K)=0.0
  800 CONTINUE
  900 CONTINUE
C
      IF(MODEX.EQ.1) RETURN
C
```

```
      WRITE (NT) B,TMASS
C
      RETURN
 1001 FORMAT (2I5,7F10.4)
 2001 FORMAT (2(3X,I4),6E15.5)
 2002 FORMAT (47H1N O D A L   L O A D S   (S T A T I C)   O R   ,
     A        29HM A S S E S   (D Y N A M I C), ///
     B        3X,4HNODE,3X,4HLOAD,
     1 2(9X,6HX-AXIS,9X,6HY-AXIS,9X,6HZ-AXIS), / 7H NUMBER,3X,4HCASE,
     2 3(10X,5HFORCE), 3(9X,6HMOMENT), / 1X)
      END
C&=========================================================================
      SUBROUTINE INPUTJ(ID,X,Y,Z,T,NUMNP,NEQ)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  MAIN
C
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6),T(1)
C
      COMMON /EXTRA/ MODEX,NT8,IFILL(14)
C
C---- SPECIAL NODE CARD FLAGS
C
C     IT    =   COORDINATE SYSTEM TYPE  (CC 1, ANY NODE CARD)
C               EQ.C, CYLINDRICAL
C     IPR   =   PRINT SUPPRESSION FLAG  (CC 6, CARD FOR NODE 1 ONLY)
C               EQ. , NORMAL PRINTING
C               EQ.A, SUPPRESS SECOND PRINTING OF NODAL ARRAY DATA
C               EQ.B, SUPPRESS PRINTING OF ID-ARRAY
C               EQ.C, BOTH *A* AND *B*
C
      DIMENSION IPRC(4)
C
      DATA IPRC/1H ,1HA,1HB,1HC/
C
      IPR = IPRC(1)
      RAD = DATAN(1.0D0)/45.0D0
C
C
C---- READ OR  GENERATE NODAL POINT DATA--------------------------------
      WRITE (33,2000)
      WRITE (33,2001)
      NOLD=0
   10 READ  (5,1000) IT,N,JPR,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N)
C
C
C**  **  **  NEXT LINE IS DELETED FOR NOT PRINTING NODAL INPUT DATA
C     WRITE (33,2002) IT,N,JPR,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N)
C**  **  **  **
C
      IF(N.EQ.1) IPR = JPR
      IF(IT.NE.IPRC(4)) GO TO 15
      DUM = Z(N)* RAD
```

```
      Z(N) = X(N)*DCOS(DUM)
      X(N) = X(N)*DSIN(DUM)
   15 CONTINUE
      IF(NOLD.EQ.0) GO TO 50
C-----CHECK IF GENERATION IS REQUIRED-----------------------------------------
      DO 20 I=1,6
      IF(ID(N,I).EQ.0.AND.ID(NOLD,I).LT.0) ID(N,I)=ID(NOLD,I)
   20 CONTINUE
      IF(KN.EQ.0) GO TO 50
      NUM=(N-NOLD)/KN
      NUMN=NUM-1
      IF(NUMN.LT.1) GO TO 50
      XNUM=NUM
      DX=(X(N)-X(NOLD))/XNUM
      DY=(Y(N)-Y(NOLD))/XNUM
      DZ=(Z(N)-Z(NOLD))/XNUM
      DT=(T(N)-T(NOLD))/XNUM
      K=NOLD
      DO 30 J=1,NUMN
      KK=K
      K=K+KN
      X(K)=X(KK)+DX
      Y(K)=Y(KK)+DY
      Z(K)=Z(KK)+DZ
      T(K)=T(KK)+DT
      DO 30 I=1,6
      ID(K,I)=ID(KK,I)
      IF (ID(K,I).GT.1) ID(K,I)=ID(KK,I)+KN
   30 CONTINUE
C
   50 NOLD=N
      IF(N.NE.NUMNP) GO TO 10
C
C---- PRINT ALL NODAL POINT DATA----------------------------------------------
C
      IF(IPR.EQ.IPRC(2) .OR. IPR.EQ.IPRC(4)) GO TO 52
      WRITE (33,2003)
      WRITE (33,2001)
      WRITE (33,2005) (N,(ID(N,I),I=1,6),X(N),Y(N),Z(N),T(N),N=1,NUMNP)
   52 CONTINUE
C
C-----NUMBER UNKNOWNS AND SET MASTER NODES NEGATIVE---------------------------
C
      NEQ=0
      DO 60 N=1,NUMNP
      DO 60 I=1,6
      ID(N,I)=IABS(ID(N,I))
      IF(ID(N,I)-1) 57,58,59
   57 NEQ=NEQ+1
      ID(N,I)=NEQ
      GO TO 60
   58 ID(N,I)=0
      GO TO 60
   59 ID(N,I)=-ID(N,I)
```

```
   60 CONTINUE
C
C---- PRINT MASTER INDEX ARRAY
C
      IF(IPR.EQ.IPRC(3) .OR. IPR.EQ.IPRC(4)) GO TO 62
      WRITE (33,2004) (N,(ID(N,I),I=1,6),N=1,NUMNP)
   62 CONTINUE
      IF(MODEX.EQ.0) GO TO 70
C***  DATA PORTHOLE SAVE
      WRITE (NT8) ((ID(N,I),I=1,6),N=1,NUMNP)
      WRITE (NT8) (X(N),N=1,NUMNP)
      WRITE (NT8) (Y(N),N=1,NUMNP)
      WRITE (NT8) (Z(N),N=1,NUMNP)
      WRITE (NT8) (T(N),N=1,NUMNP)
      ENDFILE NT8
C
      REWIND 2
      WRITE (2) ID
C
      RETURN
C
   70 CONTINUE
      REWIND 8
      WRITE (8) ID
C
      RETURN
C
 1000 FORMAT (2(A1,I4),5I5,3F10.0,I5,F10.0)
 2000 FORMAT (//23H NODAL POINT INPUT DATA )
 2001 FORMAT (5HONODE 3X 24HBOUNDARY CONDITION CODES 11X
     .  23HNODAL POINT COORDINATES / 7H NUMBER 2X 1HX 4X 1HY 4X 1HZ 3X
     . 2HXX 3X 2HYY 3X 2HZZ12X 1HX 12X 1HY 12X 1HZ 12X 1HT )
C
C
C**  **  **  NEXT LINE IS IGNORED WITH LINE #31600
C2002 FORMAT (1X,A1,I4,A1,I3,5I5,3F13.3,I5,F13.3)
C**  **  **
C
C
 2003 FORMAT (//21H1GENERATED NODAL DATA)
 2004 FORMAT (//17H1EQUATION NUMBERS/
     1 35H    N     X     Y     Z    XX    YY    ZZ /(7I5))
 2005 FORMAT (I5,6I5,4F13.3)
      END
C&=================================================================
      SUBROUTINE INP21 (NUMMAT,MAXTP,NORTHO,NDLS,NOPSET,NT8SV,NUMNP,X,
     1          Y,Z,DEN,RHO,NTP,EE,DCA,NFACE,LT,PWA,LOC,MAXPTS)
C
C     CALLED BY : THDFE
C     CALLS : VECTR2,CROSS2
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C
```

```
C     THIS ROUTINE READS AND PRINTS ALL 21-NODE SOLID ELEMENT DATA
C     BETWEEN THE CONTROL CARD AND THE ELEMENT DATA CARDS
C
C
      COMMON / JUNK/  XLE(4),YLE(4),ZLE(4),TLE(4),PLE(4),FILL1(22),V2(3)
      COMMON /EXTRA/ MODEX,NT8
C
      DIMENSION   X(1),Y(1),Z(1),DEN(1),RHO(1),NTP(1),EE(MAXTP,13,1),
     1            DCA(3,3,1),NFACE(1),LT(1),PWA(7,1),LOC(7,1),
     2            MAXPTS(1)
      DIMENSION   HED(6)
C
C     READ AND PRINT OF MATERIAL PROPERTIES
C
      WRITE (33,3000)
C
      DO 10 I=1,NUMMAT
C
      READ (5,1001) M,NTP(I),DEN(I),RHO(I),(HED(N),N=1,6)
C
C     SET DEFAULT VALUES IF REQUIRED AND CHECK FOR INPUT ERRORS
C
      IF(RHO(I).EQ.0.0) RHO(I) = DEN(I) / 386.4
      IF(NTP(I).EQ.0) NTP(I) = 1
C
      WRITE (33,3002) M,NTP(I),DEN(I),RHO(I),(HED(N),N=1,6)
C
      IF(I.EQ.M) GO TO 2
      WRITE (33,4001)
      STOP
C
    2 IF(NTP(M).LE.MAXTP) GO TO 4
      WRITE (33,4002) MAXTP
      STOP
    4 NT = NTP(M)
C
C     READ PROPERTIES FOR EACH TEMPERATURE
C
      DO 6 K=1,NT
      READ (5,1002) (EE(K,L,M),L=1,13)
      WRITE (33,3003) (EE(K,L,M),L=1,13)
    6 CONTINUE
C
C     TEMPERATURE CARDS MUST BE ASCENDING ORDER
C
      IF(NT.EQ.1) GO TO 10
      DO 8 J=2,NT
      IF(EE(J,1,M).GT.EE(J 1,1,M)) GO TO 8
      WRITE (33,4003)
      STOP
    8 CONTINUE
   10 CONTINUE
C***  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.0) GO TO 12
```

```
      DO 11 M=1,NUMMAT
      WRITE (NT8) M,NTP(M),DEN(M),RHO(M)
      NT = NTP(M)
      WRITE (NT8) ((EE(K,L,M),L=1,13),K=1,NT)
   11 CONTINUE
C***
C
C     MATERIAL AXIS ORIENTATION SETS
C
   12 IF(NORTHO.EQ.0) GO TO 21
C
      WRITE (33,3004)
C
      DO 20 M=1,NORTHO
      READ (5,1003) N,NI,NJ,NK
      WRITE (33,3005) N,NI,NJ,NK
C
C***  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.1)
     *WRITE (NT8)    N,NI,NJ,NK
C***
C     CHECK FOR ADMISSABILITY OF DATA
C
      IF(N.EQ.M) GO TO 13
      WRITE (33,4004)
      STOP
C
   13 IF(NI.GT.0 .AND. NI.LE.NUMNP) GO TO 5015
      L = NI
 5014 WRITE (33,4005) L
      STOP
 5015 IF(NJ.GT.0 .AND. NJ.LE.NUMNP) GO TO 5016
      L = NJ
      GO TO 5014
 5016 IF(NK.GT.0 .AND. NK.LE.NUMNP) GO TO 14
      L = NK
      GO TO 5014
   14 CONTINUE
C
C     GENERATE DIRECTION COSINE ARRAY FOR THIS DATA SET
C
      CALL VECTR2 (DCA(1,1,M),X(NI),Y(NI),Z(NI),X(NJ),Y(NJ),Z(NJ),IERR)
      IF(IERR.EQ.0) GO TO 16
      WRITE (33,4006)
      STOP
   16 CALL VECTR2 (V2,X(NI),Y(NI),Z(NI),X(NK),Y(NK),Z(NK),IERR)
      IF(IERR.EQ.0) GO TO 17
      WRITE (33,4007)
      STOP
   17 CALL CROSS2 (DCA(1,1,M),V2,DCA(1,3,M),IERR)
      IF(IERR.EQ.0) GO TO 18
      WRITE (33,4008)
      STOP
   18 CALL CROSS2 (DCA(1,3,M),DCA(1,1,M),DCA(1,2,M),IERR)
```

```
      IF(IERR.EQ.0) GO TO 20
      WRITE (33,4009)
      STOP
   20 CONTINUE
C
C     READ AND PRINT DISTRIBUTED SURFACE LOAD DATA
C
   21 IF(NDLS.EQ.0) GO TO 31
C
      WRITE (33,3006)
C
      DO 30 M=1,NDLS
C
      READ  (5,1004) N,NFACE(M),LT(M)
      WRITE (33,3007) N,NFACE(M),LT(M)
C
C     CHECK FOR DATA ADMISSABILITY
C
      IF(N.EQ.M) GO TO 22
      WRITE (33,4010)
      STOP
   22 IF(NFACE(M).GE.1 .AND. NFACE(M).LE.6) GO TO 23
      WRITE (33,4011)
      STOP
   23 IF(LT(M).EQ.0) LT(M) = 1
      IF(LT(M).GE.1 .AND. LT(M).LE.2) GO TO 24
      WRITE (33,4012)
      STOP
   24 IF(LT(M).EQ.2) GO TO 26
      READ (5,1005) (PWA(I,M),I=1,4)
      DO 25 I=2,4
   25 IF(PWA(I,M).EQ.0.0) PWA(I,M) = PWA(1,M)
      WRITE (33,3008) (PWA(I,M),I=1,4)
      GO TO 30
   26 READ (5,1005) (PWA(I,M),I=1,7)
      WRITE (33,3009) (PWA(I,M),I=1,7)
   30 CONTINUE
C
CAAA  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.0) GO TO 5031
      DO 5030 M=1,NDLS
      WRITE (NT8) NFACE(M),LT(M),(PWA(I,M),I=1,7)
 5030 CONTINUE
 5031 CONTINUE
CAAA
C
C     READ AND PRINT OF STRESS OUTPUT REQUEST LOCATION SETS
C
   31 IF(NOPSET.EQ.0) GO TO 49
C
      WRITE (33,3010) (I,I=1,7)
      WRITE (34,*) '---STRESS OUTPUT LOCATIONS---'
C
      DO 40 M=1,NOPSET
```

```
      BB = 0.0
      CC = 0.0
      DO 120 I=1,3
      AA = AA + XJ(MM,I)**2
      CC = CC + XJ(MN,I)**2
```



```
      READ (5,1006)  (LOC(I,M),I=1,7)
      WRITE (33,3011) M,(LOC(I,M),I=1,7)
      WRITE (34,3011) M,(LOC(I,M),I=1,7)
C
      L = 0
      DO 35 J=1,7
      IF(LOC(J,M).EQ.0) GO TO 36
      L = L + 1
      IF(LOC(J,M).GE.1 .AND. LOC(J,M).LE.27) GO TO 35
      WRITE (33,4013) J
      MODEX = 1
      GO TO 36
   35 CONTINUE
C
   36 IF(L.GT.0) GO TO 37
      L = 1
      LOC(1,M) = 21
   37 MAXPTS(M) = L
C
   40 CONTINUE
C***  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.1)
     *WRITE (NT8) ((LOC(I,J),I=1,7),J=1,NOPSET)
C***
C
C     ELEMENT LOAD CASE MULTIPLIERS
C
   49 WRITE (33,3012)
C
      READ (5,1007) XLF,YLF,ZLF,TLF,PLF
      WRITE (33,3013) XLF,YLF,ZLF,TLF,PLF
C***  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.1)
     *WRITE (NT8)   XLF,YLF,ZLF,TLF,PLF
C***
C
      RETURN
C
C     FORMATS
C
 1001 FORMAT (2I5,2F10.0,6A6)
 1002 FORMAT (7F10.0/6F10.0)
 1003 FORMAT (4I5)
 1004 FORMAT(3I5)
 1005 FORMAT (7F10.0)
 1006 FORMAT (7I5)
 1007 FORMAT (4F10.0)
C
 3000 FORMAT (//38H MATERIAL PROPERTY TABLES
 3002 FORMAT (//22HOMATERIAL NUMBER   = (,I3,1H),/
     1          10H NUMBER OF,                       /
     2          23H TEMPERATURE POINTS = (,I3,1H),/
     3          23H WEIGHT DENSITY     = (    ,E12.4,1H),/
     4          23H MASS   DENSITY     = (,E12.4,1H),/
```

```
     5          23H IDENTIFICATION     = (,6A6,1H),//
     6 1X,11HTEMPERATURE,9X,3HE11,9X,3HE22,9X,3HE33,4X,3HV12,4X,3HV13,
     7 4X,3HV23,8X,3HG12,8X,3HG13,8X,3HG23,3X,7HALPHA-1,3X,7HALPHA-2,
     8 3X,7HALPHA-3,/1X)
3003 FORMAT (E12.2,3E12.1,3F7.3,3E11.1,3E10.3)
3004 FORMAT (//50H M A T E R I A L   A X I S   O R I E N T A T I O N   ,
     1 3X,9HT A B L E     ,//
     2 28H    SET   NODE   NODE   NODE   ,/
     3 28H NUMBER     NI     NJ     NK, / 1X)
3005 FORMAT (4I7)
3006 FORMAT(//51H D I S T R I B U T E D   S U R F A C E   L O A D
     1            11HT A B L E    ,/,1X )
3007 FORMAT (//7X,27HLOAD SET NUMBER            = ,I6 /
     1          7X,27HLOAD SURFACE ELEMENT FACE = ,I6 /
     1          7X,27HLOAD TYPE CODE            = ,I6/1X)
3008 FORMAT (12H DISTRIBUTED, 11X,4HP(1),11X,4HP(2),11X,4HP(3),11X,
     1          4HP(4), / 4X,8HPRESSURE,4E15.3)
3009 FORMAT (12H HYDROSTATIC,10X,5HGAMMA,11X,4HX(S),11X,4HY(S),11X,
     1          4HZ(S),11X,4HX(N),11X,4HY(N),11X,4HZ(N), /
     2          4X,8HPRESSURE, 7E15.3)
3010 FORMAT (//51H S T R E S S   O U T P U T   R E Q U E S T   T A B L E ,
     * //
     *8H    SET ,7(2X,5HPOINT), / 8H NUMBER ,7(4X,1H(,I1,1H)),/1X)
3011 FORMAT (I8,7I7)
3012 FORMAT (///34H E L E M E N T   L O A D   C A S E          ,3X,
     1 21HM U L T I P L I E R S          ,//
     *                       31X,6HCASE A,4X,6HCASE B,4X,6HCASE C,
     2 4X,6HCASE D,/1X)
3013 FORMAT (
     1 27H X-DIRECTION GRAVITY =       ,4F10.2/
     2 27H Y-DIRECTION GRAVITY =       ,4F10.2/
     3 27H Z-DIRECTION GRAVITY =       ,4F10.2/
     4 27H THERMAL LOADING     =       ,4F10.2/
     5 27H PRESSURE LOADING    =       ,4F10.2  //1X)
C
4001 FORMAT (40H0ERROR***   MATERIAL CARDS OUT OF ORDER.,/1X)
4002 FORMAT (52H0ERROR***   NUMBER OF TEMPERATURE CARDS EXCEEDS USER,
     1 10H MAXIMUM (,I4,2H)., / 1X)
4003 FORMAT (51H0ERROR***   TEMPERATURES MUST BE INPUT IN ASCENDING  ,
     1 7H ORDER., / 1X)
4004 FORMAT (47H0ERROR***   AXIS ORIENTATION CARD OUT OF ORDER.,/1X)
4005 FORMAT (30H0ERROR***   UNDEFINED NODE NUMBER = ,I5 / 1X)
4006 FORMAT (38H0ERROR***   VECTOR IJ HAS ZERO LENGTH.,/1X)
4007 FORMAT (38H0ERROR***   VECTOR IK HAS ZERO LENGTH.,/1X)
4008 FORMAT (43H0ERROR***   IJ AND IK VECTORS ARE PARALLEL.,/1X)
4009 FORMAT (43H0ERROR***   E3 AND E1 VECTORS ARE PARALLEL.,/1X)
4010 FORMAT (50H0ERROR***   SET NUMBERS MUST BE IN ASCENDING ORDER,/1X)
4011 FORMAT (40H0ERROR***   INVALID SURFACE FACE NUMBER.,/1X)
4012 FORMAT (30H0ERROR***   INVALID LOAD TYPE.,/1X)
4013 FORMAT (42H0ERROR***   INVALID OUTPUT POINT NUMBER = , I6 / 1X)
C
C
      END
C&==================================================================
```

```
      SUBROUTINE PRINTD (ID,D,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT,MQ)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  SOLEQ,SOLEIG,RESPEC
C
      DIMENSION ID(NUMNP,6),B(NEQB,LL),D(6,LL)
      DATA Q11,Q21,Q12,Q22,Q13,Q23/'  LOAD','  CASE','EIGEN-','VECTOR',
     1     ' MODE ','NUMBER'/
C
      REWIND 8
      READ (8) ID
      M=NEQ
      NN=NEQB*NBLOCK
C
      IF (MQ.EQ.2) GO TO 50
      IF (MQ.EQ.3) GO TO 55
      REWIND NT
      Q1=Q11
      Q2=Q21
      GO TO 60
   50 Q1=Q12
      Q2=Q22
      GO TO 60
   55 Q1=Q13
      Q2=Q23
      REWIND NT
      READ (NT)
   60 continue
c--   WRITE (33,2003) Q1,Q2      !removed as there is a print in SOLEQ
C
      N=NUMNP
      rewind nt      !****************
C
      DO 500 KK=1,NUMNP
C
      I=6
      DO 250 II=1,6
      DO 100 L=1,LL
  100 D(I,L)=0.
      IF(M.GT.NN) GO TO 150
      IF (M.EQ.0) GO TO 150
      READ (NT) B
      NN=NN-NEQB
  150 IF(ID(N,I).LT.1) GO TO 250
      K=M-NN
      M=M-1
C
      DO 200 L=1,LL
  200 D(I,L)=B(K,L)
  250 I=I-1
C
cc-   WRITE (33,2004) N,(L,(D(I,L),I=1,6),L=1,LL)
C
  500 N=N-1
```

```
C
      RETURN
C
 2003 FORMAT (1H1,38HN O D E   D I S P L A C E M E N T S / ,
     1         17HR O T A T I O N S,// 3X,4HNODE,2X,A6,2(12X,2HX-,12X,
     2         2HY-,12X,2HZ-), / 7H NUMBER,2X,A6,3(3X,11HTRANSLATION),
     3         3(6X,8HROTATION), / 1X)
 2004 FORMAT (I6,I8,6E14.5,: / (7X,I8,6E14.5) )
C-- 2004 FORMAT (1H0,I6,I8,6E14.5 / (7X,I8,6E14.5) )
C
      END
C&=====================================================================
      SUBROUTINE SOL21
C
C     CALLED BY : ELTYPE
C     CALLS : STRSC
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     3 / D 8 TO 21 NODE SOLID ELEMENTS
C
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/ NS,ND,LM(63)
      COMMON/JUNK/ LT,LH,L,N6,SIG(42),N7,N8,N9,N10,N11,N12,N13,N14,
     1           N15,N16,N17
      COMMON /EXTRA/ MODEX,NT8,N10SV,NT10
C
      COMMON   A(1)
C
C
      IF(NPAR(1).EQ.0) GO TO 500
C
C     ERROR CHECKS AND SET DEFAULT VALUES IF REQUIRED
C
      WRITE (33,1000)
      IF(NPAR(2).GT.0) GO TO 10
      WRITE (33,1001) (NPAR(K),K=1,10)
      WRITE (33,1002)
      STOP
   10 IF(NPAR(3).GT.0) GO TO 20
      WRITE (33,1001) (NPAR(K),K=1,10)
      WRITE (33,1003)
      STOP
   20 IF(NPAR(4).EQ.0) NPAR(4) = 1
      IF(NPAR(7).EQ.0) NPAR(7) = 21
      IF(NPAR(7).GE.8 .AND. NPAR(7).LE.21) GO TO 30
      WRITE (33,1001) (NPAR(K),K=1,10)
      WRITE (33,1004)
      STOP
   30 IF(NPAR(9).EQ.0) NPAR(9) = 2
      IF(NPAR(9).GE.2 .AND. NPAR(9).LE.4) GO TO 40
      WRITE (33,1001) (NPAR(K),K=1,10)
      WRITE (33,1005)
      STOP
```

```
   40 IF(NPAR(10).EQ.0) NPAR(10) = 2
      IF(NPAR(10).GE.2 .AND. NPAR(10).LE.4) GO TO 50
      WRITE (33,1001) (NPAR(K),K=1,10)
      WRITE (33,1005)
      STOP
C
C     STORAGE ALLOCATION
C
C     A(N6) = STARTING LOCATION OF WEIGHT DENSITY
C     A(N7) = STARTING LOCATION OF MASS DENSITY
C     A(N8) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
C              NUMBER OF TEMPERATURE POINTS FOR EACH MATERIAL TABLE
C     A(N9) = STARTING LOCATION OF MATERIAL PROPERTY TABLE
C     A(N10) = STARTING LOCATION OF DIRECTION COSINE ARRAYS FOR
C              MATERIAL ORIENTATION AXIS
C     A(N11) = STARTING LOCATION OF SURFACE LOAD FACE NUMBERS
C     A(N12) = STARTING LOCATION OF SURFACE LOAD CODE TYPES
C     A(N13) = STARTING LOCATION OF PRESSURE WORKING ARRAY
C     A(N14) = STARTING LOCATION OF OUTPUT REQUEST LOCATION SETS
C     A(N15) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
C              NUMBER OF REQUESTED OUTPUT LOCATION IN EACH OUTPUT SET
C     A(N16) = STARTING LOCATION OF ELEMENT STIFFNESS MATRIX
C
   50 N6 = N5 + NUMNP
      N7 = N6 + NPAR(3)
      N8 = N7 + NPAR(3)
      N9 = N8 + NPAR(3)
      N10 = N9 + NPAR(3) * NPAR(4) * 13
      N11 = N10 + NPAR(5) * 9
      N12 = N11 + NPAR(6)
      N13 = N12 + NPAR(6)
      N14 = N13 + NPAR(6) * 7
      N15 = N14 + NPAR(8) * 7
      N16 = N15 + NPAR(8)
      N17 = N16 + NPAR(7) * 189
C
      IF(N17.GT.MTOT) CALL ERROR(N17-MTOT)
C
C     PROCESS ELEMENT DATA, AND GENERATE ELEMENT MATRICES
C
      CALL THDFE (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
     1           A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),
     2           NPAR(2),NPAR(3),NPAR(4),NPAR(5),NPAR(6),NPAR(7),
     3           NPAR(8),NPAR(9),NPAR(10),NUMNP)
C
      RETURN
C
C     RECOVER ELEMENT STRESSES (STATIC CASES ONLY)
C
  500 WRITE (34,2001)
      NUME = NPAR(2)
C
      read (5,*) nll,nuu
  501 format(3i5)
```

```
      if(nll.le.0) nll=1
      if(nuu.le.0) nuu=nume
      DO 800 MM=1,NUME
C
C
CAAA  STRESS PORTHOLE
      CALL STRSC (A(N1),A(N3),NEQ,0)
      IF(N10SV.EQ.1)
     *WRITE (NT10) NS
CAAA
C
      IF(NS.EQ.1) GO TO 800
C
C-    WRITE (34,5000)
C
      DO 700 L=LT,LH
C
C
      CALL STRSC (A(N1),A(N3),NEQ,1)
      LOC = NS/6
      K1 = -5
C
      DO 600 N=1,LOC
      K1 = K1 + 6
      K2 = K1 + 5
C
      if(mm.ge.nll.and.mm.le.nuu) then
      IF(N.EQ.1) WRITE (34,3001) mm,L,N,(SIG(I),I=K1,K2)
      IF(N.GT.1) WRITE (34,4001)  N,(SIG(I),I=K1,K2)
      end if
C
CAAA  STRESS PORTHOLE
      IF(N10SV.EQ.1)
     *WRITE (NT10) mm,L,N,(SIG(I),I=K1,K2)
CAAA
  600 CONTINUE
C
C-    WRITE (34,5000)
C
  700 CONTINUE
  800 CONTINUE
      RETURN
C
C     FORMATS
C
 1000 FORMAT (53H121 - N O D E  S O L I D  E L E M E N T  I N P U T   .
     1 10HD A T A   ,//38H0CONTROL INFORMATION                  ,/1X)
 1001 FORMAT (48H0ERROR DETECTED WHILE PROCESSING MASTER ELEMENT ,
     1 12HCONTROL CARD,//16X,1H(,10I5,1H),/1X)
 1002 FORMAT (32H NO 3/D SOLID ELEMENTS SPECIFIED,/1X)
 1003 FORMAT (23H NO MATERIALS REQUESTED, / 1X)
 1004 FORMAT (49H MAXIMUM NUMBER OF NODES MUST BE GE.8 .AND. LE.21, 1X)
 1005 FORMAT (42H INTEGRATION ORDER MUST BE GE.2 .AND. LE.4,/1X)
 2001 FORMAT (54H121 - N O D E  S O L I D  E L E M E N T  S T R E S
```

```
     *  //
     *23H ELEMENT  LOAD LOCATION,9X,6HSIG-XX,9X,6HSIG-YY,9X,6HSIG-ZZ,
     3 9X,6HSIG-XY,9X,6HSIG-YZ,9X,6HSIG-ZX,//1X)
 3001 FORMAT (I8,I6,I9,6E15.6)
 4001 FORMAT (  14X,I9,6E15.6)
 5000 FORMAT ( / )
C
      END
C&=====================================================================
      SUBROUTINE SSLAW (D,E,TEMP,DCA,KAXES,KMAT,NEL,DUM,ALPHA)
C
C     CALLED BY : THDFE
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     THIS ROUTINE FORMS THE STRESS-STRAIN LAW IN MATERIAL COORDINATES
C     (X1,X2,X3) AND TRANSFORMS THE MATERIAL SYSTEM LAW TO GLOBAL
C     COORDINATES (X,Y,Z).
C
      DIMENSION D(6,6),E(12),TEMP(6,6),DCA(3,3),IPRM(3),DUM(6,6),
     1          ALPHA(6)
C
      DATA IPRM / 2,3,1 /
C
C     FORM THE DIRECT STRAIN PARTITION OF THE STRAIN-STRESS LAW IN
C     MATERIAL COORDINATES (X1,X2,X3)
C
      DO 20 I=1,3
      ALPHA(I)   = E(I+9)
      ALPHA(I+3) = 0.0
      IF(E(I).GT.1.0E-08) GO TO 15
      WRITE (33,3000) I,I,KMAT,NEL
      STOP
 3000 FORMAT (23H0ERROR***   MODULUS E(,2I1,16H) FOR MATERIAL (,I3,
     1          14H) IN ELEMENT (,I5,10H) IS ZERO., / 1X)
   15 TEMP(I,I) = 1.0/E(I)
   20 CONTINUE
C
      TEMP(1,2) = -E(4)* TEMP(2,2)
      TEMP(2,1) =        TEMP(1,2)
      TEMP(1,3) = -E(5)* TEMP(3,3)
      TEMP(3,1) =        TEMP(1,3)
      TEMP(2,3) = -E(6)* TEMP(3,3)
      TEMP(3,2) =        TEMP(2,3)
C
C     INVERT THE DIRECT STRAIN PARTITION
C
      DO 60 N=1,3
      X = 1.0/TEMP(N,N)
      DO 30 J=1,3
   30 TEMP(N,J) = - TEMP(N,J)* X
C
      DO 50 I=1,3
      IF(N.EQ.I) GO TO 50
```

```
      DO 40 J=1,3
      IF(N.EQ.J) GO TO 40
      TEMP(I,J) = TEMP(I,J) + TEMP(I,N) * TEMP(N,J)
   40 CONTINUE
   50 TEMP(I,N) = TEMP(I,N) * X
C
      TEMP(N,N) = X
   60 CONTINUE
C
C     FORM THE COMPLETE STRESS-STRAIN LAW IN MATERIAL COORDINATES
C
      DO 70 I=1,6
      DO 70 J=1,6
   70 D(I,J) = 0.0
C
      DO 80 I=1,3
      DO 80 J=1,3
   80 D(I,J) = TEMP(I,J)
C
      D(4,4) =E(7)
       (5,5) = E(9)
      D(6,6) = E(8)
C
C     TRANSFORM THE MATERIAL LAW TO GLOBAL COORDINATES (X,Y,Z)
C
      IF(KAXES.LT.1) RETURN
C
C     TRANSFORMATION BETWEEN MATERIAL STRAINS AND GLOBAL STRAINS
C
      DO 100 I1=1,3
      I2 = IPRM(I1)
      DO 90 J1 = 1,3
      J2 = IPRM(J1)
      TEMP(I1  ,J1  ) = DCA(J1,I1)*DCA(J1,I1)
      TEMP(I1+3,J1  ) = DCA(J1,I1)*DCA(J1,I2) * 2.0
      TEMP(I1  ,J1+3) = DCA(J1,I1)*DCA(J2,I1)
      TEMP(I1+3,J1+3) = DCA(J1,I1)*DCA(J2,I2) +
     1                  DCA(J2,I1)*DCA(J1,I2)
   90 CONTINUE
  100 CONTINUE
C
C     ROTATE THE MATERIAL LAW TO THE GLOBAL SYSTEM
C
      DO 130 I=1,6
      DO 120 J=1,6
      X = 0.0
      DO 110 K=1,6
  110 X = X + D(I,K)*TEMP(K,J)
  120 DUM(I,J) = X
  130 CONTINUE
C
      DO 160 I=1,6
      DO 150 J=I,6
      X = 0.0
```

```
      DO 140 K=1,6
  140 X = X + TEMP(K,I)*DUM(K,J)
      D(I,J) = X
      D(J,I) = X
  150 CONTINUE
  160 CONTINUE
C
C     TRANSFORM THE EXPANSION COEFFICIENTS FROM MATERIAL COORDINATES
C     TO GLOBAL COORDINATES
C
C
      DO 200 I=1,6
      X = 0.0
      DO 190 K=1,3
  190 X = X + TEMP(K,I)*E(K+9)
      IF(I.GT.3) X =X*2.0
  200 ALPHA(I) = X
C
      RETURN
      END
C&=====================================================================
      SUBROUTINE STRESS(STR,B,D,NEQB,LB,LL,NEQ,NBLOCK)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLS:  ELTYPE
C     CALLED BY:  SOLEQ
C
      DIMENSION D(NEQ,LB),B(NEQB,LL),STR(4,LL)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
      COMMON /JUNK/ LT,LH,IFILL(428)
      COMMON /EXTRA/ MODEX,NT8,N10SV,NT10,IFILL2(12)
C
      READ (8) STR
      NT=(LL-1)/LB +1
      LH=0
C***  STRESS PORTHOLE
      IF(N10SV.EQ.1)
     *WRITE (NT10) NELTYP,NT
C
      DO 1000 II=1,NT
C
      LT =LH+1
      LLT=1-LT
      LH=LT+LB-1
      IF(LH.GT.LL) LH=LL
C
C     MOVE DISPLACEMENTS INTO CORE FOR LB LOAD CONDITIONS
C
      REWIND 2
C***  STRESS PORTHOLE
      IF(N10SV.EQ.1)
     *WRITE (NT10) LT,LH
      NQ=NEQB*NBLOCK
      DO 200 NN=1,NBLOCK
```

```
      READ (2) B
      N=NEQB
      IF (NN.EQ.1) N=NEQ-NQ+NEQB
      NQ=NQ-NEQB
      DO 200 J=1,N
      I=NQ+J
      DO 200 L=LT,LH
      K=L+LLT
  200 D(I,K)=B(J,L)
      LK=LH-LT+1
C
C     CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS
C
      REWIND 1
      DO 1000 M=1,NELTYP
      READ (1) NPAR
C***  STRESS PORTHOLE
      IF(N10SV.EQ.1)
     *WRITE (NT10) NPAR
      MTYPE=NPAR(1)
      NPAR(1)=0
      CALL ELTYPE(MTYPE)
 1000 CONTINUE
C
      RETURN
      END
C&=====================================================================
      SUBROUTINE STRSC(STR,D,NEQ,NTAG)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLED BY:  TRUSS,BEAM,PLANE,THREED,SHELL,BOUND,PIPE
C
      DIMENSION STR(4,1),D(NEQ,1)
      COMMON /JUNK/ LT,LH,L,IPAD,SG(20),SIG(7),EXTRA(186)
      COMMON /EM/ NS,ND,B(42,63),TI(42,4),LM(63)
C
      IF (NTAG.EQ.0) GO TO 800
      LL=L-LT+1
      DO 300 I=1,NS
      SG(I)=0.0
      DO 300 J=1,4
  300 SG(I)=SG(I)+TI(I,J)*STR(J,L)
      DO 500 J=1,ND
      JJ=LM(J)
      IF(JJ.EQ.0) GO TO 500
      DO 400 I=1,NS
  400 SG(I)=SG(I)+B(I,J)*D(JJ,LL)
C
  500 CONTINUE
      GO TO 900
  800 READ  (1) ND,NS,(LM(I),I=1,ND),(( B(I,J),I=1,NS),J=1,ND),
     1 ((TI(I,J),I=1,NS),J=1,4)
  900 RETURN
      END
```

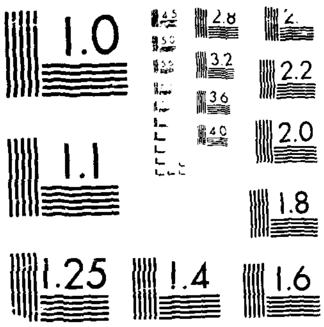
```
C&=========================================================================
      SUBROUTINE ST8R21 (E,B,S,XX,NOD9,H,P,SIGDT,DELT,FT,DL,XM,NEL,ND,
     1                   IELD,IELX,KTL,KGL,KMS,NINT,NINTZ,WTDEN,MSDEN)
C
C     CALLED BY : THREE
C     CALLS : DER3DS
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                  .
C .                                                                  .
C .      HEXAHEDRAL CURVILINEAR THREE-DIMENSIONAL ELEMENTS           .
C .                                                                  .
C .      ISOPARAMETRIC OR SUBPARAMETRIC                              .
C .                                                                  .
C .                                                                  .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C
      DIMENSION E(6,1),B(6,1),S(63,1),XX(3,1),NOD9(1),H(1),P(3,1),
     1          SIGDT(1),DELT(1),FT(1),DL(1),XM(1),D(9),SDT(6),BV(63),
     2          W(3,3),IPERM(3,3),KDX(3),LDX(3)
C
      COMMON /GAUSS/ XG(4,4),WGT(4,4)
C     REAL MSDEN
      REAL*8 MSDEN
C
      DATA IPERM / 1,4,6, 4,2,5, 6,5,3 /
C
      VOL = 0.0
C
C     DETERMINE IF THE MATERIAL IS ORTHOTROPIC (ISO.EQ.1, ISOTROPIC)
C
      DUM = 0.0
      DO 20 I=4,6
      J = I-1
      DO 20 K=1,J
   20 DUM = DUM +DABS(E(K,I))
      ISO = 1
      IF(DUM.GT.1.0E-6) ISO = 0
      IF(ISO.EQ.0) GO TO 24
      DO 22 I=2,3
      DUM = DUM +DABS(E(I  ,I  ) -E(I-1,I-1))
   22 DUM = DUM +DABS(E(I+3,I+3) -E(I+2,I+2))
      DUM = DUM +DABS(E(1  ,2 ) - E(2  ,3  ))
      DUM = DUM +DABS(E(2  ,3  ) - E(3  ,1  ))
      IF ( DUM.GT.1.0E-6 ) ISO=0
   24 CONTINUE
C
C
C     VOLUME INTEGRATION LOOP
```

```
C
C
      DO 10 LX=1,NINT
      DO 10 LY=1,NINT
      E1=XG(LX,NINT)
      E2=XG(LY,NINT)
      DO 10 LZ=1,NINTZ
      E3=XG(LZ,NINTZ)
C
      WT=WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINTZ)
C
C     EVALUATE STRAIN-DISPLACEMENT MATRIX B AND JACOBIAN DETERMINANT
C
      CALL DER3DS (NEL,XX,B,DET,E1,E2,E3,NOD9,H,P,IELD,IELX)
C
C     ADD CONTRIBUTION TO ELEMENT STIFFNESS
C
      FACT = WT* DET
      FACT2 =DSQRT(FACT)
C
      DO 25 I=1,IELD
      K3 = 3*I
      K2 = K3-1
      K1 = K2-1
      BV(K1) = B(1,K1)* FACT2
      BV(K2) = B(2,K2)* FACT2
      BV(K3) = B(3,K3)* FACT2
   25 CONTINUE
C
      DO 30 I=1,ND
      DO 30 J=I,ND
   30 S(I,J) = S(I,J) + BV(I)* BV(J)
C
C     ACCUMULATE ELEMENT VOLUME
C
      VOL = VOL + FACT
C
C     COMPUTE GRAVITY LOADS
C
      IF(KGL.EQ.0) GO TO 150
      DO 130 K=1,IELD
  130 DL(K) = DL(K) + H(K)*FACT* WTDEN
C
C     COMPUTE THERMAL LOADING NODE FORCE VECTOR
C
  150 IF(KTL.EQ.0) GO TO 190
C
C         1. ELEMENT TEMPERATURE DIFFERENCE AT THIS INTEGRATION POINT
C            (R,S,T)
C
      DT = 0.0
      DO 160 K=1,IELD
  160 DT = DT + H(K)* DELT(K)
      DT = DT* FACT
```

```
C
C        2. INITIAL STRESSES AT (R,S,T)
C
        DO 170 K=1,6
    170 SDT(K) = SIGDT(K)*DT
C
C        3. NODE FORCES
C
        DO 180 K=1,ND
        DO 175 I=1,6
    175 FT(K) = FT(K) + B(I,K)* SDT(I)
    180 CONTINUE
C
C        WRITE(28,*) ' DT,DELT',DT,(DELT(K),K=1,IELD)
C        WRITE(28,*) ' SIGDT  ',(SIGDT(K),K=1,6)
C        WRITE(28,*) '   FT -----------------------'
C        WRITE(28,*)  (FT(K),K=1,6)
    190 CONTINUE
     10 CONTINUE
C
        DO 35 I=1,2
        IC = ND-I
        DO 35 J=1,IC
        M=J+I
     35 S(M,J) = S(J,M)
C
C       COMPLETE THE K-MATRIX WITH APPROPRIATE MATERIAL CONSTANT MULT-
C       PLICATIONS OF THE INTEGRATED B(I)*B(J) ARRAY.
C
C          1. TEST FOR MATERIAL TYPE
C
        IF(ISO.EQ.0) GO TO 75
C
C             A.  ISOTROPIC MATERIAL
C
        D1 = E(1,1)
        D2 = E(1,2)
        D3 = E(4,4)
C
        DO 60 I=1,IELD
        K3 = 3*I
        K2 = K3-1
        K1 = K2-1
        K0 = K1-1
        DO 60 J=I,IELD
        L3 = 3*J
        L2 = L3-1
        L1 = L2-1
        L0 = L1-1
C
        IC = 0
        DO 40II=1,3
        M = II+ K0
        DO 40 JJ=1,3
```

```
      N = JJ+ L0
      IC = IC+ 1
      D(IC) = S(M,N)
   40 CONTINUE
C
      S(K1,L1) = D(1)* D1 + (D(5) + D(9))* D3
      S(K2,L2) = D(5)* D1 + (D(1) + D(9))* D3
      S(K3,L3) = D(9)* D1 + (D(5) + D(1))* D3
      S(K1,L2) = D(2)* D2 + D(4)* D3
      S(K2,L1) = D(4)* D2 + D(2)* D3
      S(K2,L3) = D(6)* D2 + D(8)* D3
      S(K3,L2) = D(8)* D2 + D(6)* D3
      S(K3,L1) = D(7)* D2 + D(3)* D3
      S(K1,L3) = D(3)* D2 + D(7)* D3
C
   60 CONTINUE
C
      GO TO 110
C
C           B. ANISOTROPIC MATERIAL
C
   75 DO 100 I=1,IELD
      K0 = 3*I-3
      DO 100 J=I,IELD
      L0 = 3*J-3
C
      DO 80 II=1,3
      M = II+K0
      DO 80 JJ=1,3
      N = JJ+L0
      W(II,JJ) = S(M,N)
   80 CONTINUE
C
      DO 100 K=1,3
      I1 = K0+K
      DO 82 IJ=1,3
   82 KDX(IJ)=IPERM(IJ,K)
      DO 95 L=1,3
      I2 = L0+L
      DO 83 IJ=1,3
   83 LDX(IJ)=IPERM(IJ,L)
C
      SUM=0.0
C
      DO 90 II=1,3
      K1 = KDX(II)
      DO 85 JJ=1,3
      K2 = LDX(JJ)
C
   85 SUM = SUM + W(II,JJ)*E(K1,K2)
   90 CONTINUE
C
      S(I1,I2) = SUM
C
```

AD-A192 952 A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF 3/3
NOTCHED COMPOSITE..(U) DREXEL INST OF TECH PHILADELPHIA
PA DEPT OF MECHANICAL ENGINE.. A S WANG ET AL. FEB 88
UNCLASSIFIED AFOSR-TR-88-0280 AFOSR-84-0334 F/G 11/4 NL



MICROCOPY RESOLUTION TEST CHART

```
   95 CONTINUE
  100 CONTINUE
  110 CONTINUE
C
C
C     REFLECT FOR SYMMETRY
C
      DO 200 I=1,ND
      DO 200 J=I,ND
  200 S(J,I) = S(I,J)
C
C     CONSTRUCT THE LUMPED MASS MATRIX
C
      IF(KMS.EQ.0) RETURN
C
      FACT = VOL* MSDEN/ IELD
      DO 220 K=1,ND
  220 XM(K) = FACT
C
C
      RETURN
      END
C&=======================================================================
      SUBROUTINE THDFE (ID,X,Y,Z,T,DEN,RHO,NTP,EE,
     1          DCA,NFACE,LT,PWA,LOC,MAXPTS,SS,
     2          NUME,NUMMAT,MAXTP,NORTHO,NDLS,MAXNOD,
     3          NOPSET,INTRS,INTT,NUMNP)
C
C     CALLED BY : SOL21
C     CALLS : INP21,CALBAN,SSLAW,DER3DS,ST8R21,FACEPR
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     ROUTINE FOR THE STIFFNESS, MASS AND STRESS MATRIX GENERATION
C     FOR THE 8-TO-21 NODE ISO-(OR SUB)-PARAMETRIC ORTHOTROPIC
C     HEXAHEDRON.
C
      COMMON /JUNK/ XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),FILL1(22),V2(3),
     1          FILL2(12),LS(4),KLS(4),NOD(21),NOD9M(13),KOD(21),
     2          NREAD,TAG,E(12)
      COMMON /ELPAR/IFILL3(15),MBAND
      COMMON /EM/ SDT(42,63),SF(42,4),NS,ND,LM(63)
      DIMENSION RF(63,4),XM(63),D(6,6),TEMP(6,6),DUM(6,6),
     *          ALPHA(6),XX(3,21),B(6,63),H(21),P(3,21),SIGDT(6),
     *          DELT(21),FT(63),DL(21),PL(63),LOCOP(7),VIS(6)
C
      COMMON /GAUSS/ XG(4,4),WGT(4,4),STPTS(27,3)
      COMMON /DYN / IFILL4(11),NDYN
      COMMON /EXTRA/ MODEX,NT8
C
      DIMENSION ID(NUMNP,1),X(1),Y(1),Z(1),T(1),DEN(1),RHO(1),
     1          NTP(1),EE(MAXTP,13,1),DCA(3,3,1),NFACE(1),LT(1),
     2          PWA(7,1),LOC(7,1),MAXPTS(1),SS(63,1)
C
```

```
C
      DATA  TG1, TG2  /'*', ' '/
      STPTS(1,1)=1.
      STPTS(2,1)=-1.
      STPTS(3,1)=-1.
      STPTS(4,1)=1.
      STPTS(5,1)=1.
      STPTS(6,1)=-1.
      STPTS(7,1)=-1.
      STPTS(8,1)=1.
      STPTS(9,1)=0.
      STPTS(10,1)=-1.
      STPTS(11,1)=0.
      STPTS(12,1)=1.
      STPTS(13,1)=0.
      STPTS(14,1)=-1.
      STPTS(15,1)=0.
      STPTS(16,1)=1.
      STPTS(17,1)=1.
      STPTS(18,1)=-1.
      STPTS(19,1)=-1.
      STPTS(20,1)=1.
      STPTS(21,1 =0.
      STPTS(22,1)=1.
      STPTS(23,1)=-1.
      STPTS(24,1)=0.
      STPTS(25,1)=0.
      STPTS(26,1)=0.
      STPTS(27,1)=0.
      STPTS(1,2)=1.
      STPTS(2,2)=1.
      STPTS(3,2)=-1.
      STPTS(4,2)=-1.
      STPTS(5,2)=1.
      STPTS(6,2)=1.
      STPTS(7,2)=-1.
      STPTS(8,2)=-1.
      STPTS(9,2)=1.
      STPTS(10,2)=0.
      STPTS(11,2)=-1.
      STPTS(12,2)=0.
      STPTS(13,2)=1.
      STPTS(14,2)=0.
      STPTS(15,2)=-1.
      STPTS(16,2)=0.
      STPTS(17,2)=1.
      STPTS(18,2)=1.
      STPTS(19,2)=-1.
      STPTS(20,2)=-1.
      STPTS(21,2)=0.
      STPTS(22,2)=0.
      STPTS(23,2)=0.
      STPTS(24,2)=1.
      STPTS(25,2)=-1
```

```
      STPTS(26,2)=0.
      STPTS(27,2)=0.
      STPTS( 1,3)=1.
      STPTS( 2,3)=1.
      STPTS( 3,3)=1.
      STPTS( 4,3)=1.
      STPTS( 5,3)=-1.
      STPTS( 6,3)=-1.
      STPTS( 7,3)=-1.
      STPTS( 8,3)=-1.
      STPTS( 9,3)= 1.
      STPTS(10,3)= 1.
      STPTS(11,3)= 1.
      STPTS(12,3)= 1.
      STPTS(13,3)=-1.
      STPTS(14,3)=-1.
      STPTS(15,3)=-1.
      STPTS(16,3)=-1.
      STPTS(17,3)=0.
      STPTS(18,3)=0.
      STPTS(19,3)=0.
      STPTS(20,3)=0.
      STPTS(21,3)=0.
      STPTS(22,3)=0.
      STPTS(23,3)=0.
      STPTS(24,3)=0.
      STPTS(25,3)=0.
      STPTS(26,3)=1.
      STPTS(27,3)=-1.
      XG(1,1) = 0.
      XG(2,1) = 0.
      XG(3,1) = 0.
      XG(4,1) = 0.
      XG(1,2) =     -.5773502691896D0
      XG(2,2) =      .5773502691896D0
      XG(3,2) = 0.
      XG(4,2) = 0.
      XG(1,3) =     -.7745966692415D0
      XG(2,3) = 0.
      XG(3,3) =      .7745966692415D0
      XG(4,3) = 0.
      XG(1,4) =     -.8611363115941D0
      XG(2,4) =     -.3399810435849D0
      XG(3,4) =      .3399810435849D0
      XG(4,4) =      .8611363115941D0
      WGT(1,1) = 2.0
      WGT(2,1) = 0.0
      WGT(3,1) = 0.0
      WGT(4,1) = 0.0
      WGT(1,2) = 1.0
      WGT(2,2) = 1.0
      WGT(3,2) = 0.0
      WGT(4,2) = 0.0
      WGT(1,3) = .5555555555556    D0
```

```
      WGT(2,3) = .8888888888888889   D0
      WGT(3,3) = .5555555555555556   D0
      WGT(4,3) = 0.0
      WGT(1,4) = .3478548451375   D0
      WGT(2,4) = .6521451548625   D0
      WGT(3,4) = .6521451548625   D0
      WGT(4,4) = .3478548451375   D0
C
      NT8SV = MODEX
      DO 10 I=4,6
      DO 10 J=1,63
   10 B(I,J) = 0.0
      DO 14 I=1,42
      DO 14 J=1,4
   14 SF(I,J)=0.0
C
C     PRINT ELEMENT CONTROL VARIABLES
C
      WRITE (33,3001) NUME,NUMMAT,MAXTP,NORTHO,NDLS,MAXNOD,NOPSET,INTRS,
     1          INTT
C
C     READ AND CHECK INPUT UP TO THE ELEMENT DATA CARDS
C
      CALL INP21      (NUMMAT,MAXTP,NORTHO,NDLS,NOPSET,NT8SV,NUMNP,X,
     1                 Y,Z,DEN,RHO,NTP,EE,DCA,NFACE,LT,PWA,LOC,MAXPTS)
C
C     READ ELEMENT DATA CARDS
C
                NREAD = 8
      IF(MAXNOD.GT.8) NREAD = 21
C
      WRITE (33,3014) (I,I=1,8)
      IF(MAXNOD.GT.8)
     *WRITE (33,3016) (I,I=9,21)
C
      NEL = 0
C
C     CARD FOR ELEMENT NUMBER ONE ONLY
C
      READ (5,1008) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      READ (5,1009) (NOD(I),I=1,NREAD)
      IREUSE = 0
      IF(INEL.EQ.1) GO TO 51
      WRITE (33,4014) INEL
      WRITE (33,4014)
      STOP
C
C     CARDS FOR ALL OTHER ELEMENTS
C
   50 READ (5,1008) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      READ (5,1009) (NOD(I),I=1,NREAD)
C
```

```
C     DATA ADMISSIBILITY CHECK
C
   51 IF(NDIS.EQ.0) NDIS = MAXNOD
      IF(NDIS.LE.MAXNOD) GO TO 5051
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      WRITE (33,4015) NDIS,MAXNOD
      STOP
 5051 IF(NDIS.GE.8) GO TO 52
      WRITE (33,4023) NDIS
      STOP
   52 IF(NXYZ.EQ.0) NXYZ = NDIS
      IF(NXYZ.LE.NDIS) GO TO 5052
      WRITE (33,4016) NXYZ,NDIS
      WRITE (33,4099)
      MODEX = 1
      GO TO 53
 5052 IF(NXYZ.GE.8) GO TO 53
      WRITE (33,4024) NXYZ
      WRITE (33,4099)
      MODEX = 1
   53 IF(NMAT.GE.1 .AND. NMAT.LE.NUMMAT) GO TO 54
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      WRITE (33,4017)
      WRITE (33,4099)
      MODEX = 1
   54 IF(MAXES.LE.NORTHO) GO TO 55
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      WRITE (33,4018)
      WRITE (33,4099)
      MODEX = 1
   55 IF(IOP.GE.0 .AND. IOP.LE.NOPSET) GO TO 56
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(I),I=1,4)
      WRITE (33,4019)
      WRITE (33,4099)
      MODEX = 1
   56 DO 57 I=1,4
      IF(LS(I).GE.0 .AND. LS(I).LE.NDLS) GO TO 57
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(J),J=1,4)
      WRITE (33,4020) LS(I)
      WRITE (33,4099)
      MODEX = 1
   57 CONTINUE
C
C     DEFAULT VALUES IF REQUIRED
C
      IF(KG.EQ.0) KG = 1
      IF(NRSINT.EQ.0) NRSINT = INTRS
      IF(NTINT.EQ.0) NTINT = INTT
C
```

```
      DO 58 I=1,8
      IF(NOD(I).GE.1 .AND. NOD(I).LE.NUMNP) GO TO 58
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(J),J=1,4)
      WRITE (33,4021) I,NOD(I)
      STOP
   58 CONTINUE
      IF(MAXNOD.LT.9) GO TO 60
      II = 0
      DO 59 I=9,21
      IF(NOD(I).EQ.0) GO TO 59
      II = II + 1
      NOD9M(II) = I
      IF(NOD(I).LE.NUMNP) GO TO 59
      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
     1,IREUSE,(LS(J),J=1,4)
      WRITE (33,4021) I,NOD(I)
      STOP
   59 CONTINUE
C
      I = II + 8
      IF(I.EQ.NDIS) GO TO 60
      WRITE (33,4025) I,NDIS
      STOP
C
   60 NEL = NEL + 1
      ML = INEL - NEL
      IF(ML) 65,70,80
   65 WRITE (33,4022) INEL
      STOP
C
C     SAVE THE DATA FOR ELEMENT NUMBER *INEL* FOR POSSIBLE USE IN
C     DATA GENERATION
C
C
   70 KDIS = NDIS
      KXYZ  = NXYZ
      KMAT  = NMAT
      KAXES = MAXES
      KIOP  = IOP
      TTZ   = TZ
      KKG   = KG
      KRSINT = NRSINT
      KTINT  = NTINT
      KREUSE = IREUSE
      DO 72 I=1,4
   72 KLS(I) = LS(I)
      DO 74 I=1,NREAD
   74 KOD(I) = NOD(I)
      TAG = TG1
C
      GO TO 90
C
C     INCREMENT THE NON-ZERO NODE NUMBERS FROM THE PRECEEDING ELEMENT
```

```
C       DATA ADMISSIBILITY CHECK
C
     51 IF(NDIS.EQ.0) NDIS = MAXNOD
```

```
C
     60 DO 85 I=1,NREAD
        IF(KOD(I).LT.1) GO TO 85
        KOD(I) = KOD(I) + KKG
     85 CONTINUE
        TAG = TG2
C
     90 ND = 3 * NDIS
C
C       COMPUTE THE AVERAGE ELEMENT TEMPERATURE USING COORDINATE NODES
C
        TAV = 0.0
        DO 95 K=1,KXYZ
        I = KOD(K)
     95 TAV = TAV + T(I)
        TAV = TAV / KXYZ
C
C       PERFORM TEMPERATURE INTERPOLATION FOR THE PROPERTY SET
C
        NT = NTP(KMAT)
        IF(NT.GT.1) GO TO 100
     97 DO 98 I=1,12
     98 E(I) = EE(1,I+1,KMAT)
        GO TO 112
    100 IF(TAV.GE.EE(1,1,KMAT)) GO TO 104
    102 WRITE (33,4030) TAV,NEL,KMAT
        STOP
    104 IF(TAV.GT.EE(NT,1,KMAT)) GO TO 102
        IF(TAV.EQ.EE(1,1,KMAT)) GO TO 97
C
        IF(MODEX.EQ.1) GO TO 112
C
        DO 106 K=2,NT
        K2 = K
        K1 = K-1
        IF(TAV.GT.EE(K1,1,KMAT) .AND. TAV.LE.EE(K2,1,KMAT)) GO TO 108
    106 CONTINUE
    108 DT = EE(K2,1,KMAT) - EE(K1,1,KMAT)
        RATIO = (TAV - EE(K1,1,KMAT)) / DT
        DO 110 I=1,12
    110 E(I) = EE(K1,I+1,KMAT) + RATIO *(EE(K2,I+1,KMAT)-EE(K1,I+1,KMAT))
C
    112 CONTINUE
C
C       FORM THE STRESS-STRAIN LAW IN MATERIAL COORDINATES AND TRANSFORM
C       TO GLOBAL (X,Y,Z) COORDINATES
C
        IF(MODEX.EQ.0)
       *CALL SSLAW (D,E,TEMP,DCA(1,1,KAXES),KAXES,KMAT,NEL,DUM,ALPHA)
C
C       STORE THE NODE COORDINATES FOR THIS ELEMENT
C
        IF(MODEX.EQ.1) GO TO 410
C
```

```
      DO 130 I=1,KDIS
      II = KOD(I)
      IF(I.LT.9) GO TO 125
      JJ = NOD9M(I-8)
      II = KOD(JJ)
  125 XX(1,I) = X(II)
      XX(2,I) = Y(II)
      XX(3,I) = Z(II)
  130 CONTINUE
C
C     COMPUTE THE ELEMENT STIFFENESS, MASS, THERMAL AND GRAVITY LOAD
C     MATRICES
C
      DO 170 I=1,63
      DO 170 J=1,4
  170 RF(I,J)=0.0
C
      IF(KREUSE.EQ.1) GO TO 300
C
      DO 180 I=1,KDIS
  180 DL(I)=0.0
      DO 190 I=1,ND
C
C
C        1. THERMAL LOADS
C
  190 FT(I)=0.0
      KTL = 0
      DUX = 0.0
      DO 200 I=1,4
  200 DUX = DUX +DABS(TLF(I))
      IF(DUX.GT.1.0E-06) KTL = 1
      IF(KTL.EQ.1) THEN    !!!!!
      WRITE(99,*) '$$$$$$$$$$ ktl===1'      !!!!!
      END IF   !!!!!
      IF (NDYN.GT.0) KTL=0
      IF(KTL.EQ.0 .OR. NDYN.GT.0) GO TO 235
C
C           A.INITIAL STRESS CONSTANTS
C
      DO 210 I=1,6
      SIGDT(I) = 0.0
      DO 205 K=1,6
  205 SIGDT(I) = SIGDT(I) + D(I,K)* ALPHA(K) ! 1 changed to I (first)
  210 CONTINUE
C      WRITE(28,*)  '  sigdt in THDEE'
C      WRITE(28,*)  (sigdt(k),k=1,6)
C
C           B. VECTOR OF NODE TEMPERATURE DIFFERENCES
C
      DO 230 I=1,KDIS
      II = KOD(I)
      IF(I.LT.9) GO TO 220
      J = NOD9M(I-8)
```

```
      II = KOD(J)
  220 DELT(I) = T(II) - TTZ
  230 CONTINUE
C
C           C. CLEAR THE THERMAL LOAD NODE FORCE VECTOR
C
C         2. GRAVITY LOADS
C
  235 DUX=0.0
      DO 250 I=1,4
  250 DUX = DUX +DABS(XLE(I)) +DABS(YLE(I)) +DABS(ZLE(I))
      KGL = 0
      IF(DUX.GT.1.0E-6) KGL = 1
      IF (NDYN.GT.0) KGL=0
C
C
C         3. MASS MATRIX
      KMS = 0
      IF(NDYN.GT.0) KMS = 1
C
      DO 270 K=1,ND
C
C         4. STIFFNESS MATRIX
C
  270 XM(K) = 0.0
      DO 280 I=1,ND
      DO 280 K=I,ND
  280 SS(I,K) = 0.0
C
C
      CALL ST8R21 (D,B,SS,XX,NOD9M,H,P,SIGDT,DELT,ET,DL,XM,NEL,ND,KDIS,
     1         KXYZ,KTL,KGL,KMS,KRSINT,KTINT,DEN(KMAT),RHO(KMAT))
C
C
C     NODE FORCES DUE TO THERMAL DISTORTION
C
  300 IF (KTL.EQ.0) GO TO 325
      DO 320 I=1,ND
      DO 310 K=1,4
  310 RE(I,K) = ET(I)* TLE(K)
  320 CONTINUE
C
C     NODE FORCES DUE TO STATIC ACCELERATIONS
C
C
  325 IF (KGL.EQ.0) GO TO 350
      DO 340 I=1,KDIS
      K3 = 3*I
      K2 = K3-1
      K1 = K2-1
      DO 330 L=1,4
      RE(K1,L) = RE(K1,L) + XLE(L)*DL(I)
      RE(K2,L) = RE(K2,L) + YLE(L)* DL(I)
  330 RE(K3,L) = RE(K3,L) + ZLE(L)* DL(I)
```

```
  340 CONTINUE
C
C     COMPUTE NODE FORCES DUE TO ELEMENT SURFACE LOADINGS
C
  350 IF(NDLS.LT.1.OR.NDYN.GT.0) GO TO 405
C
      DO 400 L=1,4
      IF(PLE(L).EQ.0.0) GO TO 400
      M = KLS(L)
      IF(M.LT.1) GO TO 400
      DO 360 K=1,ND
C
  360 PL(K) = 0.0
      CALL FACEPR (NEL,KDIS,KXYZ,XX,NOD9M,H,P,PL,NFACE(M),LT(M),
     1            PWA(1,M),M)
C
      DO 370 I=1,ND
C
  370 RE(I,L) = RE(I,L) + PL(I)* PLE(L)
  400 CONTINUE
  405 CONTINUE
C
C     ASSIGN EQUATION NUMBERS TO THE ELEMENT DEGREES OF FREEDOM
C
  410 K = -3
      DO 420 I=1,KDIS
      II = KOD(I)
      IF(I.LT.9) GO TO 415
      JJ = NOD9M(I-8)
      II = KOD(JJ)
  415 K = K+3
      DO 420 L=1,3
      M = K+L
  420 LM(M) = ID(II,L)
C
      IF(KIOP.GT.0) NS = 6*MAXPTS(KIOP)
      IF(KIOP.EQ.0) NS = 6
      IF (NDYN.GT.0) NS=42
C
C     SAVE STIFFNESS AND LOAD MATRICES
C
      CALL CALBAN (MBAND,NDIF,LM,XM,SS,RF,ND,63,NS)
C
C     COMPUTE STRESS RECOVERY MATRICES
C
      IF (NDYN.LT.1) GO TO 425
      NOP=7
      DO 422 I=1,7
  422 LOCOP(I)=I + 20
      GO TO 450
  425 IF (KIOP.EQ.0) GO TO 440
      NOP = MAXPTS(KIOP)
      DO 430 I=1,NOP
  430 LOCOP(I) = LOC(I,KIOP)
```

```
      GO TO 450
  440 NOP = 1
      LOCOP(1) = 21
C
  450 IF(MODEX.EQ.1) GO TO 510
C
C     CONSIDER EACH OUTPUT LOCATION
C
      DO 500 L=1,NOP
C
      M= LOCOP(L)
      E1= STPTS(M,1)
      E2= STPTS(M,2)
      E3= STPTS(M,3)
C
C     COMPUTE THE STRAIN-DISPLACEMENT MATRIX AT THIS LOCATION
C
      CALL DER3DS (NEL,XX,B,DET,E1,E2,E3,NOD9M,H,P,KDIS,KXYZ)
C
      DO 470 I=1,6
      N= 6*(L-1)+I
      DO 465 J=1,ND
      Q = 0.0
      DO 460 K=1,6
  460 Q = Q + D(I,K)* B(K,J)
  465 SDT(N,J) = Q
  470 CONTINUE
C
C     FORM THE INITIAL STRESS CORRECTIONS DUE TO THERMAL LOADS
C
      IF(KTL.EQ.0 .OR. NDYN.GT.0) GO TO 500
C
C
C        1. TEMPERATURE DIFFERENCE AT THIS LOCATION
C
      Q = 0.0
      DO 480 K=1,KDIS
C
C        2. VECTOR OF INITIAL STRESSES
C
  480 Q = Q + H(K)* DELT(K)
      DO 485 K=1,6
  485 VIS(K) = -Q * SIGDT(K)
C
      DO 490 I=1,6
      N = 6*(L-1)+I
C
      DO 490 K=1,4
  490 SE(N,K) = VIS(I)* TLE(K)
C
  500 CONTINUE
C
C     SAVE THE STRESS RECOVERY ARRAYS
C
```

```
C
  510 CONTINUE
C
      IF(MODEX.EQ.0)
     1WRITE (1) ND,NS,(LM(I),I=1,ND),((SDT(I,J),I=1,NS),J=1,ND),
     2          ((SF(I,J),I=1,NS),J=1,4)
C
C     PRINT DATA FOR THE CURRENT ELEMENT
C
      WRITE (33,3015) NEL,KDIS,KXYZ,KMAT,KAXES,KIOP,TTZ,NNG,KRSINT,KTINT,
     1          KREUSE,KLS
      WRITE (33,3017) (KOD(I),I=1,NREAD)
C
C***  DATA PORTHOLE SAVE
      IF(NT8SV.EQ.1)
     1WRITE (NT8)   NEL,KDIS,KXYZ,KMAT,KAXES,KIOP,TTZ,   KRSINT,KTINT,
     2          KREUSE,KLS,NREAD,
     3          (KOD(I),I=1,NREAD)
C***
C
C     CHECK FOR THE LAST ELEMENT
C
      IF(NUME-NEL) 65,600,530
  530 IF(ML)       50, 50, 60
C
  600 RETURN
C
C     FORMATS
C
 1008 FORMAT (6I5,F10.0,4I5,4I2)
 1009 FORMAT (16I5)
C
 3001 FORMAT ( 7X,34HNUMBER OF 21-NODE ELEMENTS        = I6//
     1          7X,34HNUMBER OF MATERIAL SETS          = I6//
     2          7X,26HMAXIMUM NUMBER OF MATERIAL,      /
     3          7X,34HTEMPERATURE INPUT POINTS         = I6 //
     4          7X,19HNUMBER OF MATERIAL,              /
     5          7X,34HAXIS ORIENTATION SETS            = I6//
     *          7X,34HNUMBER OF DISTRIBUTED LOAD SETS  = I6//
     6          7X,34HMAXIMUM NUMBER OF ELEMENT NODES  = I6 //
     7          7X,34HNUMBER OF STRESS OUTPUT SETS     = I6 //
     8          7X,34HR,S COORDINATE INTEGRATION ORDER = I6 //
     9          7X,34HT COORDINATE INTEGRATION ORDER   =I6 // 1X
 3014 FORMAT (52H13 / D   8   T O   2 1   N O D E   S O L I D   E L E
     1  18H M E N T   D A T A, //  8H ELEMENT 2(2X,5HNODES),2(2X,
     2  5HMATL.),2X,6HSTRESS,4X,6HSTRESS,2X,4HNODE,2(2X,5HGAUSS),6X,
     3  2HK-,5X,3HLSA,3X,3HLSB,3X,3HLSC,3X,3HLSD,  /
     4  8H  NUMBER,7H -NDIS-,7H -NXYZ-,2X,5HTABLE,3X,4HAXES,2X,6HOUTPUT,
     5  6X,4HFREE,2X,4HINC.,2(3X,4HPTS.),2X,6HMATRIX,2X,4(2X,4H-OR-), /
     6  26X,3HNO.,4X,3HSET,5X,3HSET,5X,5HTEMP.,2X,4H-KG-,2X,5H-R,S-,4X,
     7  3H-T-,2X,6HRE-USE,2X, 8(2X,2HN-,I2)  )
 3015 FORMAT (I8,4I7,I8,F10.1,I6,2I7,I8,2X,4I6)
 3016 FORMAT (84X,8(2X,2HN-,I2),: / 84X,5(2X,2HN-,I2) )
 3017 FORMAT (84X,8I6,: / 84X,8I6,:/ 84X,5I6)
```

```
C
 4014 FORMAT (33HOERRORAAA   ENCOUNTERED ELEMENT (,I5,13H), BUT EXPECT,
     1          21H TO READ ELEMENT ONE., / 1X)
 4015 FORMAT (42HOERRORAAA   NUMBER OF DISPLACEMENT NODES (,I5,4H) IS,
     1          30H LARGER THAN MAXIMUM ALLOWED (,I5,2H)., / 1X)
 4016 FORMAT (40HOERRORAAA   NUMBER OF COORDINATE NODES (,I5,6H) MUST,
     1          39H BE .LE. NUMBER OF DISPLACEMENT NODES (,I5,2H).)
 4017 FORMAT (36HOERRORAAA   ILLEGAL MATERIAL NUMBER. )
 4018 FORMAT (44HOERRORAAA   ILLEGAL MATERIAL AXIS REFERENCE.  )
 4019 FORMAT (41HOERRORAAA   ILLEGAL OUTPUT SET REFRENCE.  )
 4020 FORMAT (41HOERRORAAA   PRESSURE LOAD SET REFERENCE (,I5,4H) IS,
     1          9H ILLEGAL. )
 4021 FORMAT (16HOERRORAAA   THE ,I2,18H-TH ELEMENT NODE (,I5,4H) IS,
     1          9H ILLEGAL.,/ 1X)
 4022 FORMAT (28HOERRORAAA   ELEMENT NUMBER (,I5,11H) IS OUT OF,
     1          10H SEQUENCE., / 1X)
 4023 FORMAT (42HOERRORAAA   NUMBER OF DISPLACEMENT NODES (,I5,
     1          25H) MUST BE AT LEAST EIGHT. )
 4024 FORMAT (40HOERRORAAA   NUMBER OF COORDINATE NODES (,I5,
     1          25H) MUST BE AT LEAST EIGHT. )
 4025 FORMAT (38HOERRORAAA   NUMBER OF NON-ZERO NODES (,I3,6H) READ,
     1          50H DOES NOT EQUAL THE NUMBER OF DISPLACEMENT NODES (,
     2          I3,2H).,/ 1X)
 4030 FORMAT (33HOERRORAAA   AVERAGE TEMPERATURE (,F10.2,5H) FOR,
     1          10H ELEMENT (,I5,29H) OUT OF RANGE FOR MATERIAL (,I3,
     2          2H)., / 1X)
 4099 FORMAT (12X,31HPROCEED IN DATA CHECK ONLY MODE, / 1X)
C
      END
C&=====================================================================
      SUBROUTINE VECTR2 (V,XI,YI,ZI,XJ,YJ,ZJ,IERR)
C
C     CALLED BY : INP21
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     THIS ROUTINE FORMS A UNIT LENGTH VECTOR *V* FROM POINT *I*
C     TO POINT *J* IN X,Y,Z SPACE
C
      DIMENSION V(3)
C
      IERR = 1
      X = XJ - XI
      Y = YJ - YI
      Z = ZJ - ZI
      XLN =DSQRT(X*X+Y*Y+Z*Z)
      IF(XLN.LE.1.0E-08) RETURN
      XLN = 1.0 / XLN
      IERR = 0
      V(3) = Z * XLN
      V(2) = Y * XLN
      V(1) = X * XLN
      RETURN
      END
```

```
C&=========================================================================
      SUBROUTINE STIME
      TS=0.0
      RETURN
      END
C
C
C&=========================================================================
C     SUBROUTINE TTIME
C         T - CUMULATIVE TASK TIME, RETURNED IN UNITS OF SECONDS
      SUBROUTINE TTIME(T)
      INTEGER*4 get_time, time
      DATA get_time /2/
      CALL LIB$STAT_TIMER(get_time,time)
      T = time / 100.0
      RETURN
      END
C&=========================================================================

      SUBROUTINE SOLEQ
      IMPLICIT REAL*8(A-H,O-Z)
C
C     CALLS:  SESOL,PRINTD,STRESS
C     CALLED BY:  MAIN
C
C     STATIC SOLUTION PHASE
C
      COMMON A(1)                !   ????????
      COMMON /ELPAR/ NP(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /SOL   / NBLOCK,NEQB,LL,NF,IFILL(7)
      DIMENSION ID(3000,6),B(40000),NX(3,200)
      COMMON/CRK/ NCRKD,ICR(9000)
      DATA NCRKD/4/             !   ?????????

!****  MIN. DIMENSION = ( nb + 6*npair)  : also check in SESOL routine
      DIMENSION TMATM(600,600)
      COMMON/TMT/TMAT(600,600),TCOL(600),TCOL2(600),TCOLM(600),
     .IST(600),R(9000)

      dimension nbc(200),ifix(200),dsave(200),npc(2,200),b(6),
     . FORCNB(3,200),FORC1(3,200),FORC2(3,200),ENERG(3)

C
      REAL*4 TI(4),ttsub(10)
      INTPR= 100          !INTERMEDIATE PRINTING

      rewind 8
      read (8) ((id(n,i),n=1,numnp),i=1,6)     ! degrees of freedom
c     WRITE(23,*)  '*** id - array in soleq ********---'
c     do n=1,numnp
c     WRITE(23,1028) (id(n,i),i=1,6)
c     end do
 1028 format (2x,3015)

```

```
***   TO read data regarding
c       nodes where forces to be found whose disp. are specified  (NB)
c     & double nodes along the crack propagation    ( NPAIR)
      read (5,*)  nb  ! no of boundary nodes where forces are to found
      IF(INTPR.LE.2) WRITE(33,*)  '    nb - ',nb
c --  input node, degree of freedom, displacement
      do ij=1,nb
      read(5,*)  nbc(ij),ifix(ij),dsave(ij)
!      if(INTPR.LE.2)
       WRITE(33,*) nbc(ij),ifix(ij),dsave(ij)
      end do
      ncrkd=0
      if(INTPR.LE.2)WRITE(33,*)'node,ix,ndof,ncrkd,icr(ndof),ist(ncrkd)'
      if(INTPR.LE.2)WRITE(33,*) (ij ,(id(ij,ix),ix=1,6),ij=1,169)

      do ib=1,nb
      node=nbc(ib)
      ix=ifix(ib)
      ndof=id(node,ix)
      ncrkd=ncrkd+1
      if(INTPR.LE.2)
     . WRITE(33,1029),node,ix,ndof,ncrkd
      icr(ndof)=ncrkd
      ist(ncrkd)=-ncrkd
      if(INTPR.LE.2)
     . WRITE(33,1029),node,ix,ndof,ncrkd,icr(ndof),ist(ncrkd)
      end do
 1029 format (2x,5i8,5i8)
      if(INTPR.LE.1) WRITE(33,1028) (ICR(IJ),IJ=1,NEQ)
      if (INTPR.LE.2)  WRITE(33,1028) IST
 1030 FORMAT (2x,20E6.3)


      read (5,*) npair     ! no of double nodes & the double nodes
!      if(INTPR.LE.2)
      WRITE(33,*)    '  npair   = ',npair
      do ipair=1,npair
      read(5,*)  (npc(i,ipair),i=1,2),(NX(IJ,ipair),IJ=1,3)
!      if(INTPR.LE.2)
      WRITE(33,*)  (npc(i,ipair),i=1,2),(nx(ij,ipair),ij=1,3)
      do ix=1,2
      node=npc(ix,ipair)
      do iy=1,3            ! d.o.f. at each node
      ndof=id(node,iy)
      if(ndof.ne.0.and.icr(ndof).eq.0.and.nx(iy,ipair).eq.1)  then
      ncrkd=ncrkd+1
      icr(ndof)=ncrkd
      ist(ncrkd)=ncrkd
      end if
      end do
      end do
      end do
```

```
C
C      SOLVE FOR THE DISPLACEMENT VECTORS
C
       CALL TTImE(TT(1))

C-     N1=1
C-     NCRKD - TOTAL NO. OF ADDITIONAL COLUMNS
       LL=1+NCRKD

       NSBO=(mBAND+1 )*NEQB
       NSB=(mBAND+LL)*NEQB
C      N3=NSB+1
       N3=N3+LL*NEQB      !??????
       NSBB=NEQB*LL*(2+(mBAND-1)/NEQB)
       IF(NSBB.LT.NSB) NSBB=NSB
       N4=N3+NSBB
       mI = MBAND + NEQB -1
       if(INTPR.LE.2)
      .WRITE(33,*) ' LL  mBAND   NSB  N3  NSBB  N4    MI N1 N2 N3'
       if(INTPR.LE.2)
      .WRITE(33,10301)  LL,mBAND,NSB,N3,NSBB,N4,MI ,N1,N2,N3
10301    format(2x,12i8)
       rewind 15
       rewind 4
       do ij=1,nblock
       read (4) (A(IK),IK=1,NSBO)
       WRITE(15) (a(IJK),IJK=1,NSBO)
       end do
       CALL SESOL (A(N1),A(N3),A(N4),LL,NBLOCK,NEQB,NSB,mI,4,3,2,55)
       CALL TTImE(TT(2))
       NL=2
       NL1=18
       NWV=LL*NEQB
       REWIND NL
       REWIND NL1
       DO NJ=1,NBLOCK
       READ (NL) (A(IJ),IJ=1,NWV)
       WRITE(NL1) (A(IJ),IJ=1,NWV)
       END DO

        WRITE(16) TMAT,tcol
        do ipair=1,npair
        node1=npc(1,ipair)
        node2=npc(2,ipair)
        do idf=1,6
        ndof1=id(node1,idf)
        ndof2=id(node2,idf)
        if(ndof1.ne.0.and.ndof2.ne.0) then
        icr1=icr(ndof1)
        icr2=icr(ndof2)
        if(ist(icr1).gt.0.and.ist(icr2).gt.0) ist(icr2)=ist(icr1)
        end if
        end do
        end do
```

```
      write (33,*) ' ---       NODES      ----     D.O.F. RELEASED ---'
      do itr=1,100
      call ttime(ttsub(1))
      read (5,*) ip1,ip2,ix
      write (33,*) ip1,ip2,ix
      if(ip1.eq.9999.and.ip2.eq.9999) go to 1995
      do while (ip1.ne.0)
      if (ip2.eq.0) then
       node=ip1
       ndof=id(node,ix)
       if(ndof.ne.0) then
       icr1=icr(ndof)
       ist(icr1)=icr1
       end if
       else
       node1=ip1
       node2=ip2
       idf=ix
       ndof1=id(node1,idf)
       ndof2=id(node2,idf)
      if(ndof1.ne.0.and.ndof2.ne.0) then
       icr1=icr(ndof1)
       icr2=icr(ndof2)
       ist(icr2)=icr2
      end if
      end if
      read (5,*) ip1,ip2,ix
      write (33,*) ip1,ip2,ix
       end do

       if(INTPR.LE.2) then
       WRITE(33,*)  ' ----- IST ----'
       WRITE(33,1028) (ist(ij),ij=1,ncrkd)
       end if

       REWIND 16
       READ (16) TMAT,tcol

       do i=1,neq
       icri=icr(i)
       isti=ist(icri)
       if(icri.le.nb.and.isti.gt.0) tcol(icri)=tcol(icri)+tcol2(icri)
       end do

       do i=1,ncrkd
       isti=ist(i)
       do j=1,ncrkd
       istj=ist(j)
       if(isti.gt.0.and.istj.le.0) tcol(i)=tcol(i)-TMAT(i,j)*dsave(j) !disp
       if(isti.le.0.or.istj.le.0) TMAT(i,j)=0.
       if(isti.le.0.and.istj.le.0.AND.isti.eq.istj) TMAT(i,j)=1.
       end do
```

```
      end do
      IF(INTPR.LE.1) WRITE(33,*)
     . ' ---- TMAT,TCOL  after IST manipulation-----'
      DO I=1,NCRKD
      IF(INTPR.LE.1) WRITE(33,1091) (TMAT(I,J),J=1,NCRKD),TCOL(I)
      END DO
      do i=1,ncrkd
      tcolm(i)=0.
      do j=1,ncrkd
      tmatm(i,j)=0.
      end do
      end do
      do i=1,ncrkd
      isti=iabs(ist(i))
      tcolm(isti)=tcolm(isti)+tcol(i)
      do j=1,ncrkd
      istj=iabs(ist(j))
      tmatm(isti,istj)=tmatm(isti,istj)+TMAT(i,j)
      end do
      end do
      do i=1,ncrkd
      if(tmatm(i,i).eq.0) tmatm(i,i)=1.0
      end do
      IF(INTPR.LE.1) WRITE(33,*) ' ---- TMATm,TCOLm  before matin-----'
      DO I=1,NCRKD
      IF(INTPR.LE.1) WRITE(33,1091) (TMATm(I,J),J=1,NCRKD),TCOLm(I)
      end do

      call ttime(ttsub(2))

      call MATIN(tmatm,Ncrkd,tcolm,1,DETERM)

      call ttime(ttsub(3))
      IF(INTPR.LE.1) WRITE(33,*) ' ---- TMATm,TCOLm  after matin-----'
      DO I=1,NCRKD
      IF(INTPR.LE.1) WRITE(33,1091) (TMATm(I,J),J=1,NCRKD),TCOLm(I)
      end do
      do i=1,ncrkd
      isti=iabs(ist(i))
      tcol(i)=tcolm(isti)
      if(ist(i).le.0) tcol(i)=dsave(i)     ! disp
      end do
     NWV=LL*NEQB
      Rewind nll
      do nj=1,nblock-1
      read (nll) (a(ij),ij=1,nwv)
      end do
      do nj=1,nblock
      nconst=(nj-1)*neqb
      read (nll) (a(ij),ij=1,nwv)
      backspace nll
      backspace nll
      do i=1,neqb
      b(i+nconst)=a(i)
```

```
      do k=1,ncrkd
      nk=neqb+(k-1)*neqb+i
      b(i+nconst)=b(i+nconst)-a(nk)*tcol(k)
      end do
      end do
      IF(INTPR.LE.1) WRITE(33,*) '  --- intermediate solution ---'
      IF(INTPR.LE.1) WRITE(33,1091) (b(ij),ij=1,neq)
      end do !nj
      do i=1,neq
      ici=icr(i)
      if(ici.ne.0) b(i)=tcol(ici)
      end do
c      WRITE(33,*) '  ---  final displacement solution ----
c      WRITE(33,1091) (b(ij),ij=1,neq)
      REWIND NL
      DO NJ=NBLOCK,1,-1
      NCONST=(NJ-1)*NEQB+1
      NU=NCONST+NEQB-1
      WRITE(NL) (B(IJ),IJ=NCONST,NU)
      END DO
        call ttime(ttsub(4))
      do i=1,neq
      r(i)=0.
      end do
      rewind 15
      do nj=1,nblock
      DO IJK=NSBO,NSB
      A(IJK)=0.0
      END DO
      read (15) (a(IJK),IJK=1,NSBO)
      nconst=(nj-1)*neqb
      ij=0
      J1=1+NCONST
      do i=1,neqb
      ij=ij+1
      in=i+nconst
      r(in)=r(in)+a(ij)*b(in)
      end do
      do j=2,MBAND
      do i=1,neqb
      in=i+nconst
      jn=j+nconst+i-1
      ij=ij+1
      r(in)=r(in)+a(ij)*b(jn)
      r(jn)=r(jn)+a(ij)*b(in)
      end do
      end do
      end do
c      WRITE(33,*) ' ------ r vector ----'
c      WRITE(33,1091) (r(ij),ij=1,neq)
```

```
C
C       Correction for thermal case  --
c       To find the mechanical loads subtract the thermal loads
c       from R(*)
c       ( It is assumed no external loads are applied at double nodes)
        Rewind 15
        nsbl=neqb*mband
        do nj=1,nblock
        read (15) (a(ij),ij=1,nsbo)
        nconst=(nj-1)*neqb
        do ij=1,neqb
        r(ij+nconst)=r(ij+nconst)-a(nsbl+ij)
        end do
        end do
       N=NUMNP
       ITR1=ITR-1
       WRITE(33,582) ITR1
  582  FORMAT (1h1, ' $$$$$    STEP # ',I4, '  $$$$$$$'/1X,40(1H_))

       WRITE(33,*)  '       *****************************************'
       WRITE(33,*)  ' ---- NODAL DISPLACEMENTS AND FORCES IN SOL20---'
       WRITE(33,*)  '       *****************************************'
 CC    WRITE(33,*)  ' (no mech. loads at the double nodes)'
       WRITE(33,20034)
       WRITE(33,20035)
 20034 FORMAT(//2X,75(1H-))

       NAUX=1
       DO 500 N=1,NUMNP
       IFLAG=0
       DO 250 I=1,3
       D(I)=0.
       D(I+3)=0.0
   150 IF(ID(N,I).LT.1) GO TO 250
       IDNI=ID(N,I)
       IF(icr(IDNI).NE.0) IFLAG=1
   200 D(I)=B(NAUX)
       D(I+3)=R(NAUX)
       NAUX=NAUX+1
   250 continue
C
C      IF(INTPR.LE.3.AND.IFLAG.EQ.0) GO TO 500
       WRITE (33,2004) N,(D(I),I=1,6)
C
   500 CONTINUE

 1091  format (2x,6g12.5)
 2004  FORMAT(2X,I5,6G12.5)
       WRITE(33,20034)
 20035 FORMAT(2X,' NODE',6X,'U',11X,'V',11X,'W',10X,'Fx',10x,'Fy',
      . 10x,'Fz'/2x,75(1H-))

        IF(ITR.NE.1) THEN
```

```
      ENERG(1)=0.0
      ENERG(2)=0.0
      ENERG(3)=0.0
      DO IPAIR=1,NPAIR
      NP1=NPC(1,IPAIR)
      NP2=NPC(2,IPAIR)
      DO IDG=1,3
      ND1=ID(NP1,IDG)
      ND2=ID(NP2,IDG)
C-    WRITE(33,*) 'NP1,NP2,ND1,ND2',NP1,NP2,ND1,ND2
C-    WRITE(33,*) 'FORC1,B s' ,FORC1(IDG,IPAIR),B(ND1),B(ND2)
      naux=nx(idg,ipair)
      IF(ND1.GT.0.AND.ND2.GT.0.and.naux.ne.0)
     . ENERG(IDG)=ENERG(IDG)-(FORC2(IDG,IPAIR)*B(ND2)+
     . FORC1(IDG,IPAIR)*B(ND1))*0.50
      END DO
      END DO


      DO IB=1,NB
      NPO=NBC(IB)
c     DO IDG=1,3
      idg=ifix(ib)
      NDO=ID(NPO,IDG)
      IF(NDO.GT.0) ENERG(IDG)=ENERG(IDG)-FORCNB(IDG,IB)*B(NDO)*0.50
c     END DO
      END DO

      WRITE(34,*)
     . ' ------  ENERGY RELEASED in ( x, y, z ) directions -----'
      WRITE(34,1048) ENERG
      IF (ITR.EQ.2) WRITE(19,1049)
      WRITE(19,1048) ENERG
 1048 format (5x,3(g15.8,3x))
 1049 format (5x,7X,'X',18X,'Y',18X,'Z')
      WRITE(34,*) ' =================================================='
      END IF
      DO IPAIR=1,NPAIR
      NP1=NPC(1,IPAIR)
      NP2=NPC(2,IPAIR)
      DO IDG=1,3
      ND1=ID(NP1,IDG)
      ND2=ID(NP2,IDG)
      FORC1(IDG,IPAIR)=R(ND1)
      FORC2(IDG,IPAIR)=R(ND2)
C-    WRITE(33,*) 'NP1,NP2,ND1,ND2' , NP1,NP2,ND1,ND2
C-    WRITE(33,*) ' IDG,IPAIR, FORC1,FORC2',IDG,IPAIR,FORC1(IDG,IPAIR),
C-   . FORC2(IDG,IPAIR)
      END DO
      END DO

      DO IB=1,NB
      NPO=NBC(IB)
      DO IDG=1,3
```

```
      NDO=ID(NPO,IDG)
      FORCNB(IDG,IB)=R(NDO)
      END DO
      END DO

      call ttime(ttsub(5))

C     PRINT DISPLACEMENTS
C
      N2=N1+NUMNP*6
      N3=N2+6*LL

      LL1=1              !**********     REASSIGNED    *******

      CALL PRINTD (A(N1),A(N2),A(N3),NEQB,NUMNP,LL1,NBLOCK,NEQ,2,1)
      CALL TTIME(TT(3))
C
C     COMPUTE AND PRINT ELEMENT STRESSES
C
      N2=N1+4*LL1
      N3=N2+NEQB*LL1
      LB=(MTOT-N3)/(NEQ +12)
      CALL STRESS(A(N1),A(N2),A(N3),NEQB,LB,LL1,NEQ,NBLOCK)

C     COMPUTE TIME LOG FOR THE DOUBLE NODES SOLUTION PHASE
C
      DO K=1,4
      ttsub(K) = ttsub(K+1)-ttsub(K)
      end do
      IF(INTPR.LE.2) WRITE (33,1985) (ttsub(L),L=1,4)
1985   format(5x,' time for tmatm formation         =',f8.2,/
     .         5x,' time for matin                   =',f8.2,/
     .         5x,' time to find global disp.        =',f8.2/
     .         5x,' time to find global nodal forces =',f8.2//)
C
      end do      !itr
1995  continue
      CALL TTIME(TT(4))
C
C     COMPUTE TIME LOG FOR THE STATIC SOLUTION PHASE
C
      DO 50 K=1,3
50    TT(K) = TT(K+1)-TT(K)
      WRITE (34,2000) (TT(L),L=1,3)
C
 2000 FORMAT (///,48H S T A T I C   S O L U T I O N   T I M E   L O G,
     1         //,5X,31HEQUATION SOLUTION   =, F8.2 /
     2            5X,31HDISPLACEMENT OUTPUT =, F8.2 /
     3            5X,31HSTRESS RECOVERY     =, F8.2 //)
C
C     RETURN
      RETURN
      END
C$=====================================================================
```

```
      SUBROUTINE SESOL
     .(A,B,MAXA,NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR)
      IMPLICIT REAL*8(A-H,O-Z)

      real*4 tt(10)
C     CALLED BY:  SOLEQ

      COMMON /ELPAR/ NF(14),NUMNP,MA,NELTYP,NZ1,NZ2,NZ3,NZ4,N5,MTOT,NEQ
       COMMON/CRK/NCRKD,ICR(9000)          !  NEQ
c-     DATA ICR/0,1,2,0,0,3,0,4/          !change CVT,CVT2 line also
c-     DATA IST/-1,-2,-3,+4/         ! IST(NCRKD)
c-     DATA DISP/2.0,3.0,2.0,8.0,6*0.0/            !DISP(NEQ)
      COmmON/TMT/TmAT(600,600),TCOL(600),TCOL2(600),TCOLM(600),
     .IST(600),R(9000)
       DIMENSION A(NAV),B(NAV),mAXA(MI)
      call ttime(tt(1))
      INTPR=100
      if(INTPR.LE.2)WRITE(33,*)'NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR'
      if(INTPR.LE.2)WRITE(33,1029),NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR
 1029  format(2x,12i8)

      if(INTPR.LE.2) WRITE(33,1028) (ICR(IJ),IJ=1,NEQ)
      if(INTPR.LE.2)  WRITE(33,1028)  (IST(IJK),IJK=1,NCRKD)
 1028 FORMAT( 2X,20I4)
      if(INTPR.LE.2)  WRITE(33,1030) DISP
 1030 FORMAT (2X,20F6.3)
       MM=1
       MA2=MA - 2
       IF(MA2.EQ.0) mA2=1
       INC=NEQB - 1
       NWA=NEQB*MA
       NTB=(MA-2)/NEQB + 1
       NEB=NTB*NEQB
       NEBT=NEB + NEQB
       NWV=NEQB*NV
       NWVV=NEBT*NV

       N1=NL
       N2=NR
       if(INTPR.LE.2)
     .WRITE(33, *),' mm,ma2,inc,neqb,nwa,ntb,neb,nebt,nwv,nwvv'
       if(INTPR.LE.2)
     .WRITE(33, 1029), mm,ma2,inc,neqb,nwa,ntb,neb,nebt,nwv,nwvv
       REWIND NSTIF
       REWIND NRED
       REWIND N1
       REWIND N2

       if(INTPR.LE.2)  WRITE(33,*) ' NAV =',NAV          !***
       DO IJ=1,NAV                       !***
       A(IJ)=0.                          !***
       B(IJ)=0.                          !***
       END DO                            !***
```

```
*** Taking the npair coeffts. out and placing in a vertcial matrix
C-  NCRKD - TOTAL NO. OF ADDITIONAL COLUMNS
    IF(INTPR.LE.1) WRITE(33,*)' IB,JB,I,J,ICI,ICJ,IJ,IICJ,J2ICI,A(IJ)'
    NO=NAV-NCRKD*NEQB
    DO I=1,mI
    DO J=1,NCRKD
    IJ=I+(J-1)*mI
    B(IJ)=0.
    END DO
    END DO
    DO NJ=1,NBLOCK

    DO J=1,NCRKD
    IJ=(J-1)*mI
    INJ=NEQB+(J-1)*mI
    DO I=1,mA-1
    IJ=IJ+1
    INJ=INJ+1
    B(IJ)=B(INJ)
    END DO
    END DO

    do j=1,ncrkd
    ij=ma+(j-1)*mi
    do i=ma,mi
    b(ij)=0.
    ij=ij+1
    end do
    end do


    READ (NSTIF) (A(IJ),IJ=1,NO)
    NCONST=NEQB*(NJ-1)
    DO IB=1,NEQB
    DO JB=1,mA
    IJ=(JB-1)*NEQB+IB
    I=IB+NCONST
    J=JB+NCONST+IB-1
    IF(I.LE.NEQ.AND.J.LE.NEQ) THEN
    ICI=ICR(I)
    ICJ=ICR(J)
    J2=JB+IB-1
    J2ICI=J2+(ICI-1)*mI
    IBICJ=IB+(ICJ-1)*mI
    IF(ICI.NE.0) B(J2ICI)=A(IJ)
    IF(ICJ.NE.0) B(IBICJ)=A(IJ)

    IF(ICI.NE.0.AND.ICJ.NE.0) THEN
    TMAT(ICI,ICJ)=A(IJ)
    TMAT(ICJ,ICI)=A(IJ)
    J2ICI=J2+(ICI-1)*mI
    B(J2ICI)=0.
```

```
      IBICJ=IB+(ICJ-1)*mI
      B(IBICJ)=0.
      END IF

      IF(INTPR.LE.1) WRITE(33,1018) IB,JB,I,J,ICI,ICJ,IJ,IICJ,JICI,A(IJ)
 1018 FORMAT(2X,9I5,E10.3)
      IF(ICJ.NE.0.OR.ICI.NE.0) THEN
      A(IJ)=0.
      IF(I.EQ.J) A(IJ)=1.
      END IF
      END IF
      END DO
      END DO

      nij=neqb*mb
      do i=1,neqb
      nij=nij+1
      ici=icr(i+nconst)
      if (ici.ne.0) then
      tcol(ici)=a(nij)
      a(nij)=0.
      end if
      end do

      NIJ=NO
      DO J=1,NCRKD
      DO I=1,NEQB
      NIJ=NIJ+1
      IJ=I+(J-1)*mI
      A(NIJ)=B(IJ)
      END DO
      END DO

c-      WRITE(33,*) 'NJ=',NJ,'     Reordered  A -MATRIX'
      IF(INTPR.LE.1) then
      DO I=1,NEQB
      IA=(NJ-1)*NEQB+I
      WRITE(33,1019) (A(IJ),IJ=I,NAV,NEQB)
 1019 FORMAT(2X,10E9.2)
      END DO
      end if


      WRITE(N1) A

      END DO   !NJ - loop

      REWIND N1
      REWIND NSTIF
      DO NJ=1,NBLOCK
      READ (N1) A
      WRITE(NSTIF) A
      END DO

```

```
      IF(INTPR.LE.1) then
      WRITE(33,*)   ---- TMAT -----
      DO I=1,NCRKD
      WRITE(33,1019) (TMAT(I,J),J=1,NCRKD),TCOL(I)
      END DO
      end if
C--        STOP
     call ttime(t2)
****   main loop over all blocks
 99    REWIND NSTIF
       DO 600 NJ=1,NBLOCK
c-        WRITE(33,*)   ------ MAIN LOOP -------- NJ = ,NJ
       IF (NJ.NE.1) GO TO 10
       READ (NSTIF) (A(IJ),IJ=1,NAV)
       IF(INTPR.LE.1) WRITE(33,*) 'NJ=',NJ,'     A -MATRIX'
       IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
  1020 FORMAT(2X,11G11.4)
       IF(NEQ.GT.1) GO TO 100
       MAXA(1)=1
      WRITE(NRED) A,MAXA
      IF(A(1)) 1,174,3
    1 KK=1
C--        IF(INTPR.LE.1) WRITE(33,1010) KK,A(1)
    3 DO 5 L=1,NV
    5 A(1+L)=A(1+L)/A(1)
      KK=1+NV
      WRITE(NL) (A(KK),KK=2,KK)
      RETURN
  10   IF (NTB.EQ.1) GO TO 100
       REWIND N1
       REWIND N2
       READ (N1) (A(IJ),IJ=1,NAV)
       IF(INTPR.LE.1) WRITE(33,*) ' NJ=',NJ,'      A - MATR'
       IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
  100   continue
       IF(INTPR.LE.1) WRITE(33,*) ' BEFORE FINDING COLUMN HEIGHTS
       IMX=NEQB*(MA+NV)
       IF(INTPR.LE.1)WRITE(33,*) 'NEQB,MA,NV,IMX',NEQB,MA,NV,IMX
       DO I=1,NEQB
       IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=I,IMX,NEQB)
       IMX=IMX+1
       END DO
****   Find column heights
       KU=1
       KM=MIN0(MA,NEQB)
       MAXA(1)=1
```

```
        DO 110 N=2,MI
        IF (N.LE.MA) THEN                         !***
        KU=KU + NEQB
        KK=KU
        MM=MINO(N,KM)
        ELSE                                       !***
        KU=KU + 1
        KK=KU
        IF (N.LE.NEQB) GO TO 140
        MM=MM - 1
        END IF                                        !***
 140    DO 160 K=1,MM
        IF (A(KK)) 110,160,110
 160    KK=KK - INC
 110    MAXA(N)=KK

        IF (A(1)) 172,174,176
 174    KK=(NJ-1)*NEQB + 1
        IF (KK.GT.NEQ) GO TO 590
        IF(INTPR.LE.1) WRITE (33,1000) KK
       STOP
 172    KK=(NJ-1)*NEQB + 1
C--        IF(INTPR.LE.1) WRITE (33,1010) KK,A(1)


****    Factorize leading block

 176    DO 200 N=2,NEQB
        NH=MAXA(N)
        IF (NH-N) 200,200,210
 210    KL=N + INC
        K=N
        D=0.
        DO 220 KK=KL,NH,INC
        K=K - 1
        AKK=A(KK)
        C=AKK/A(K)
        D=D + C*AKK
 220    A(KK)=C
        A(N)=A(N) - D

        IF (A(N)) 222,224,230
 224    KK=(NJ-1)*NEQB + N
        IF (KK.GT.NEQ) GO TO 590
        IF(INTPR.LE.1) WRITE (33,1000) KK
        STOP
 222    KK=(NJ-1)*NEQB + N
C--        IF(INTPR.LE.1) WRITE (33,1010) KK,A(N)

 230    IC=NEQB
        DO 240 J=1,MA2
        MJ=MAXA(N+J) - IC
        IF (MJ.LE.N) GO TO 240          !***
        KU=MINO(MJ,NH)                 !***
```

```
        KN=N + IC
        C=0.
        DO 300 KK=KL,KU,INC
  300   C=C + A(KK)*A(KK+IC)
        A(KN)=A(KN) - C
  240   IC=IC + NEQB

        K=N + NWA
        DO 430 L=1,NV
        KJ=K
        C=0.
        DO 440 KK=KL,KH,INC
        KJ=KJ - 1
  440   C=C + A(KK)*A(KJ)
        A(K)=A(K) - C
  430   K=K + NEQB

  200   CONTINUE
        IF(INTPR.LE.1) WRITE(33,*) ' -- AFTER FACORIZING LEADING BLOCK
        IMX=NEQB*(MA+NV)
        DO I=1,NEQB
C-      IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=I,IMX,NEQB)
        IMX=IMX+1
        END DO
 1005   FORMAT(2X,10G9.2)


****    Carry over into trailing blocks

        DO 400 NK=1,NTB
        IF(INTPR.LE.1) WRITE(33,*) 'NJ,NK ',NJ,NK,'    B- MAT'
        IF ((NK+NJ).GT.NBLOCK) GO TO 400
        NI=N1
        IF ((NJ.EQ.1).OR.(NK.EQ.NTB)) NI=NSTIF
        READ (NI) (B(IJ),IJ=1,NAV)


C-      IF(INTPR.LE.1) WRITE(33,1020) (B(IJ),IJ=1,NAV)
        ML=NK*NEQB + 1
        MR=MIN0((NK+1)*NEQB,MI)
       IF(MA.EQ.1) ML=MR
        MD=MI - ML
        KL=NEQB + (NK-1)*NEQB*NEQB
        N=1

        DO 500 M=ML,MR
        NH=MAXA(M)
        KL=KL + NEQB
        IF (NH.LT.KL) GO TO 505          !***
        K=NEQB
        D=0.
        DO 520 KK=KL,NH,INC
        C=A(KK)/A(K)
        D=D + C*A(KK)
```

```
      A(KK)=C
520   K=K - 1
      B(N)=B(N) - D
      IF (MD.LE.0) GO TO 580          !***
      IC=NEQB
      DO 540 J=1,MD
      MJ=MAXA(M+J) - IC
      IF (MJ.LT.KL) GO TO 540         !***
      KU=MINO(MJ,NH)
      KN=N + IC
      C=0.
      DO 575 KK=KL,KU,INC
575   C=C + A(KK)*A(KK+IC)
      B(KN)=B(KN) - C
540   IC=IC + NEQB

580   KN=N + NWA
      K=NEQB + NWA
      DO 610 L=1,NV
      KJ=K
      C=0.
      DO 620 KK=KL,NH,INC
      C=C + A(KK)*A(KJ)
620   KJ=KJ - 1
      B(KN)=B(KN) - C
      KN=KN + NEQB
610   K=K + NEQB

505   MD=MD - 1
500   N=N + 1

      IF (NTB.NE.1) GO TO 560
      WRITE (NRED) A,MAXA
      DO 570 I=1,NAV
570   A(I)=B(I)
      GO TO 600
560   WRITE (N2) B

400   CONTINUE

      M=N1
      N1=N2
      N2=M
590   WRITE (NRED) A,MAXA

600   CONTINUE

     call ttime(tt(3))

****  Vector back-substitution

      DO 700 K=1,NWVV
700   B(K)=0.
      REWIND NL
```

```
      DO 800 NJ=1,NBLOCK
      BACKSPACE NRED

      READ (NRED) (A(IJ),IJ=1,NAV),(MAXA(IJ),IJ=1,MI)
c--   WRITE(33,*) ' Vector back sub. NJ=',NJ,'    A- MAT'

C-    IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
      BACKSPACE NRED
      K=NEBT
      DO 810 L=1,NV
      DO 820 I=1,NEB
      B(K)=B(K-NEQB)
  820 K=K - 1
  810 K=K + NEBT + NEB
      KN=0
      KK=NWA
      NDIF=NEQB
      IF (NJ.EQ.1) NDIF=NEQB - (NBLOCK*NEQB - NEQ)
      DO 855 L=1,NV
      DO 850 K=1,NDIF
  850 B(KN+K)=A(KK+K)/A(K)
      KK=KK + NEQB
  855 KN=KN + NEBT
      IF(MA.EQ.1) GO TO 915
      ML=NEQB + 1
      KL=NEQB
      DO 860 M=ML,MI
      KL=KL + NEQB
      KU=MAXA(M)
      IF (KU-KL) 860,870,870
  870 K=NEQB
      KM=M
      DO 880 L=1,NV
      KJ=K
      DO 890 KK=KL,KU,INC
      B(KJ)=B(KJ) - A(KK)*B(KM)
  890 KJ=KJ - 1
      KM=KM + NEBT
  880 K=K + NEBT
  860 CONTINUE
      N=NEQB
      DO 910 I=2,NEQB
      KL=N + INC
      KU=MAXA(N)
      IF (KU-KL) 910,920,920
  920 K=N
      DO 930 L=1,NV
      KJ=K
      DO 940 KK=KL,KU,INC
      KJ=KJ - 1
  940 B(KJ)=B(KJ) - A(KK)*B(K)
  930 K=K + NEBT
  910 N=N - 1
```

```
  915 KK=0
      KN=0
      DO 950 L=1,NV
      DO 960 K=1,NEQB
      KK=KK + 1
  960 A(KK)=B(KN+K)
  950 KN=KN + NEBT

      WRITE (NL) (A(K),K=1,NWV)
      IF(INTPR.LE.1) WRITE(33,*) '  Solution --'
      IF(INTPR.LE.1) WRITE (33,1020) (A(K),K=1,NWV)
  800 CONTINUE

      call ttime(tt(4))

****  To find y - vector
      do i=1,ncrkd
      tcol2(i)=tcol(i)
      if(ist(i).le.0) tcol(i)=0.
      end do
      rewind nstif
      backspace nl
      nc2=neqb*ma
      do nj=1,nblock

      read (nstif) a
      nij=no



      read (nl) (B(ij),ij=1,nwv)
      backspace nl
      backspace nl
      nij=neqb*(nv-ncrkd)


      NIJ10=NO
      NIJ20=NEQB*(NV-NCRKD)

      do J=1,ncrkd
      do I=1,j
      taux=0.0
      NIJO=NIJ10+(I-1)*NEQB
      NIJ=NIJ20+(J-1)*NEQB
      do k=1,neqb
      NIJO=NIJO+1
      NIJ=NIJ+1
      taux=taux-A(NIJO)*B(NIJ)
      end do
      tmat(i,j)=tmat(i,j)+taux
      end do
      end do
      do j=1,ncrkd
```

```
         do i=1,j-1
         tmat(j,i)=tmat(i,j)
         end do
         end do

         do i=1,ncrkd
         taux=0.0
         NIJ0=NIJ10+(I-1)*NEQB
         do k=1,neqb
         NIJ0=NIJ0+1
         taux=taux-A(NIJ0)*B(K)
         end do
         tcol(i)=tcol(i)+taux
         end do
         IF(INTPR.LE.1) then
         WRITE(33,*) ' ---- TMAT,TCOL  towards end -----'
         DO I=1,NCRKD
         WRITE(33,1091) (TMAT(I,J),J=1,NCRKD),TCOL(I)
 1091    format (2x,6g12.5)
         END DO
        end if

         end do   ! nj
         call ttime(tt(5))
         if(INTPR.LE.2)  WRITE(33,*) '  TIME LOG IN SESOL --------'
         do k=1,4
         tt(k)=tt(k+1)-tt(k)
         end do
         if(INTPR.LE.2)  WRITE(33,995) (tt(l),l=1,4)
 995     format(10x,'Time to form matrix for double nodes etc.=',f8.2,
     .        /10x,'Time to decompose A - matrix             =',f8.2,
     .        /10x,'Time for vector back substitution        =',f8.2,
     .        /10x,'Time to form TMAT                        =',f8.2)
  1000 FORMAT (// 46H   STOP  ***  ZERO DIAGONAL ENCOUNTERED DURING,
     1             18H EQUATION SOLUTION, /
     2        15X,18H EQUATION NUMBER =, I6 )
  1010 FORMAT (/  50H WARNING ***  NEGATIVE DIAGONAL ENCOUNTERED DURING,
     1             18H EQUATION SOLUTION, /
     2        15X,18H EQUATION NUMBER =, I6, 5X, 7HVALUE =, E20.8 )

          RETURN
          END
 C*===================================================================
        SUBROUTINE MATIN(A,N,B,M,DETERM)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION A(600,600),B(600,1),IPIVOT(600),INDEX(600,2),BT(600)
        EQUIVALENCE (IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)
        DETERM=1.0
        do 20 J=1,N
 20     IPIVOT(J)=0.0
        do 550 I=1,N
        AMAX=0.0

        do 105 J=1,N
```

```
      IF(IPIVOT(J)-1) 60,105,60
60    do 100 K=1,N
      IF(IPIVOT(K)-1) 80,100,740
80    IF(AMAX -DABS(A(J,K))) 85,100,100
85    IROW=J
      ICOLUM=K
      AMAX=DABS(A(J,K))
100   CONTINUE
105   CONTINUE

      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
      IF(IROW-ICOLUM) 140,260,140
140   DETERM=-DETERM
      do 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
200   A(ICOLUM,L)=SWAP
      IF(M) 260,260,210
210    do 250 L=1,m
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
250   B(ICOLUM,L)=SWAP
260   INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT=A(ICOLUM,ICOLUM)
      DT(I)=PIVOT
      A(ICOLUM,ICOLUM)=1.0
      do 350 L=1,N
350   A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
      IF(M) 380,380,360
360    do 370 L=1,m
370   B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
380   do 550 L1=1,N
      IF(L1-ICOLUM) 400,550,400
400   T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      do 450 L=1,N
 450   A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF(M) 550,550,460
460   do 500 L=1,M
500   B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
550   CONTINUE
      do 710 I=1,N
      L=N+1-I
C--   DETERM=DETERM*DT(L)
      IF(INDEX(L,1)-INDEX(L,2)) 630,710,630
630   JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      do 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
705   CONTINUE
710   CONTINUE
```

```
      do 11 K=1,N
      IF(IPIVOT(K).NE.1) GO TO 12
11    CONTINUE
      RETURN
12    WRITE(33,991)
991   FORMAT(10X,'MATRIX IS DSINGULAR        ')
740   RETURN
      END
```

APPENDIX - C

LISTING OF THE POSTPROCESSOR, 'PLOT'

```
!-----------------------------------------------------------------
!--     THIS IS PROGRAM FOR PLOTTING 3-D GRAPHS USING TEMPLATE
!--     ROUTINES. FROM THE OUT OF KSAP II, THIS PROGRAM CAN
!--     SORT OUT STRESSES AND CORRESPONDING COORDINATE LOCATIONS.
!--     THE STRESSES MAY BE SCALED CONVENIENTLY AND EYE
!--     COORDINATES CAN BE CHOSEN TO OBTAIN DIFFERENT SIZES OF
!--     THE SAME 3-D PLOT.
!-----------------------------------------------------------------
       DIMENSION XX(40),YY(40),Z(40,40),STRESS(15,15,6,10),
      .WORK(3200),STRE(1600),PENS(6),WIV(8)
       DIMENSION HEAD(7)
        CHARACTER*12 FILNAM
        INTEGER S13

       DATA IO/5/,IOD/7/,CURE/3.0/,OUTFIL/8.0/,FONTFIL/11.0/
       DATA PENS/1.2.3.1.1.1/
       DATA IORTHO,FPD,SX,SY/1.30.0,8.0,8.0/

       WRITE (5,*)'ENTER FILENAME'
       READ (5,555).FILNAM
 555   FORMAT (A)
       OPEN(UNIT=IOD,FILE=FILNAM,STATUS='OLD')
       WRITE(IO,*)'0:SCREEN.1:PRINTER,2:PLOTTER'
       READ(IO,*)IDEV
       READ (IOD,22),(HEAD(I),I=1.6)
 22    FORMAT(1X,6A4)

       READ (IOD,*) NNODES,NLOC
       IF (NLOC.NE.1)CALL SORT21(RAD,IOD,XX,YY,NX,NY,STRE,nnodes,nloc)
       IF (NNODES.EQ.21) GO TO 899
       if (nloc.ne.1) go to 899
       READ(IOD,*)NX,NY,NL
 !     WRITE(5,*)NX,NY,NL
       READ(IOD,*)(XX(I),I=1,NX)
 !     WRITE(5,*)(XX(I),I=1,NX)
       READ(IOD,*)(YY(I),I=1,NY)
 !     WRITE(5,*)(YY(I),I=1,NY)
       nx=nx-1
       ny=ny-1
       READ(IOD,7000)((((STRESS(I,J,K,L),K=1,6),I=1,NX),J=1,NY),L=1,NL)
 7000  FORMAT(25X,6E15.6)
 !     WRITE(5,7000)((((STRESS(I,J,K,L),K=1,6),I=1,NX),
 !     .J=1,NY),L=1,NL)
 20    WRITE(IO,*)'ENTER STRESS & LAYER #'
       READ(IO,*)S13,LN
 !     WRITE(5,*)S13,LN
       K=0
       DO J=1,NY
       DO I=1.NX
       K=K+1
       STRE(K)=STRESS(I,J,S13,LN)
        END DO
        END DO
```

```
c--     finding stress location coordinates..
c--     from nodal coordinates for 8 node element.
              do i=1,nx
        if (i.gt.1) xx(i)=(xx(i)+xx(i-1))/2.
        end do
        xx(nx)=xx(nx+1)
        do i=1,ny
        if (i.gt.1) yy(i)=(yy(i)+yy(i-1))/2.
        end do
        yy(ny)=yy(ny+1)

  899   CONTINUE
        NXNY=NX*NY
        SMAX=-1.0E+30
        SMIN=1.0E+30
        DO I=1,NXNY
        IF (STRE(I).GT.SMAX) SMAX=STRE(I)
        IF (STRE(I).LE.SMIN) SMIN=STRE(I)
        END DO
        WRITE(5,*)'SMIN= ',SMIN,' SMAX= ',SMAX
   10   WRITE(IO,*)'EYEX,EYEY,EYEZ'
        READ(IO,*)EX,EY,EZ
        WRITE(IO,*)'TYPE SCALE FACTOR'
        READ(IO,*) FACT
        DO I=1,NXNY
        STRE(I)=STRE(I)/FACT
        END DO
        DO J=1,NY
        I1=1+(J-1)*NX
        I2=I1+NX-1
        II=0
        DO I=I1,I2
        II=II+1
        Z(II,J)=STRE(I)
        END DO
C        WRITE (85,*) (STRE(I),I=I1,I2)
C        WRITE (85,*) '-------------------------------------------------
        END DO
        DO J=1,NY
C        WRITE(5,*)J,':',(Z(I,J),I=1,NX)
        END DO
        WRITE(5,7000) (STRE(I),I=1,NXNY)
C        WRITE (5,*) 'BEFORE SCALING..'
C--     SCALING THE COORDINATES----
C        CMAX=XX(NX)
C        IF (CMAX.LT.YY(NY)) CMAX=YY(NY)
C        WRITE (5,*) 'CMAX',CMAX
C       CMAX=8.0/CMAX
C       DO 444 I=1,NX
C444     XX(I)=XX(I)*CMAX
C         DO 445 J=1,NY
C445     YY(I)=YY(I)*CMAX
        FX=NX
```

```
      FY=NY
      IF (IDEV.EQ.2) CALL UCONIG(3.0)

      CALL USTART
      IF(IDEV.EQ.1) CALL UPSET('OUTF',OUTFIL)
      IF (IDEV.EQ.0) CALL UERASE
      CALL UPSET ('ENTFILE',ENTFIL)
C      CALL USET ('NOSAXIS')
      CALL USET ('ZAXIS')
      CALL USET ('SHIFT
      CALL USET ('PERC
      CALL USET ('ORIG
C      CALL USTUD (WIV
      CALL UVWPRT (0.0,100.., 0.0,100.0)
      CALL UPVSRF (SIZE,XX,YY,WORK,EX,EY,EX,EY,EZ,PENS)
      WRITE (5,*) '...plot finished..
666   IF (IDEV.EQ.0) CALL UPAUSE
7     FORMAT (A1)
      CALL UEND
999   STOP
      END

        subroutine sort21 (KAD,IOD,XP,YP,NX,NY,ZP,nnodes,nloc)
C-------------------------------------------------------------------
C--     THE FOLLOWING ARE THE STRESS LOCATIONS USED IN KSAP II
C--     PROGRAM. DEPENDING ON THE REQUIRED STRESS PLANE LOCATION
C--     SORTING WILL BE DONE. AVERAGING IS ALSO CARRIED OUT
C--     BETWEEN ADJACENT ELEMENTS.
C--     LOC - STRESS OUTPUT LOCATIONS IN SAP....
C--
c--      6--13-- 5     18--24--17     2-- 9-- 1     y
c--      !   !   !     !   !   !      !   !   !     |
c--     14--27--16     23--21--22    10--26--12     |
c--      !   !   !     !   !   !      !   !   !     ----> x
c--      7--15-- 8     19--25--20     3--11-- 4
c--       bottom         middle          top
c-------------------------------------------------------------------
        DIMENSION  XP(1),YP(1),ZP(1),LOC(7)
      ..,X(50),Y(50),Z(50),IEMT(3,3,3),ID(3),JD(3),STRES(300,7,6)
        DATA IEMT/7,19,3,14,23,10,6,18,2, 15,25,11,27,21,26,
      . 13,24,9, 8,20,4,16,22,12,5,17,1/
        WRITE (5,*) 'ENTER STRESS NO.(1,2,3,4,5,6)'
        READ (5,*) NST
        READ (IOD,*) NONX,NONY,NONZ
        READ (IOD,*) (LOC(I),I=1,NLOC)
        READ (IOD,*) (X(I),I=1,NONX)
        READ (IOD,*) (Y(I),I=1,NONY)
        READ (IOD,*) (Z(I),I=1,NONZ)
        IF (NNODES.EQ.8) THEN
        NONX=2*NONX-1
        NONY=2*NONY-1
        NONZ=2*NONZ-1
        DO 111 I=NONX,1,-1
        J=I/2
```

```
      IF (MOD(I,2).NE.0) X(I)=X(J+1)
      IF (MOD(I,2).EQ.0) X(I)=(X(I+1)+X(J))/2.
111   CONTINUE
      DO 222 I=NONY,1,-1
      J=I/2
      IF (MOD(I,2).NE.0) Y(I)=Y(J+1)
222   IF (MOD(I,2).EQ.0) Y(I)=(Y(I+1)+Y(J))/2.
      DO 333 I=NONZ,1,-1
      J=I/2
      IF (MOD(I,2).NE.0) Z(I)=Z(J+1)
333   IF (MOD(I,2).EQ.0) Z(I)=(Z(I+1)+Z(J))/2.
      END IF

      WRITE (5,*) ' SELECT LEVEL (z-coord. No.) of xy-plane'
      DO 41 I=1,NONZ
41    write (5,51) I,Z(I)
51    FORMAT (2X,I2,2X,F8.4)
      READ (5,*) nlev
      JS=0
      IF (MOD(NLEV,2).NE.0) GO TO 510
C--   MIDDLE.......
      JS=2
      DO 20 J=1,3
      DO 20 I=1,3
      IF (LOC(1).EQ.IBMT(2,J,I)) GO TO 21
20    CONTINUE
      GO TO 26
510   IF (NLEV.EQ.1) GO TO 23
      JS=3
      DO 10 J=1,3
      DO 10 I=1,3
      IF (LOC(1).EQ.IBMT(3,J,I)) GO TO 21
10    CONTINUE
      IF (NLEV.EQ.NONZ) GO TO 26
23    JS=1
      DO 11 J=1,3
      DO 11 I=1,3
      IF (LOC(1).EQ.IBMT(1,J,I)) GO TO 21
11    CONTINUE
26    WRITE (5,*) '..LEVEL NO. DOESN T MATCH WITH LOC.
     . NOS.
      STOP
21    CONTINUE
      DO 40 J=1,3
      ID(J)=0
      DO 50 I=1,3
      DO 50 K=1,NLOC
      IF (LOC(K).EQ.IBMT(JS,I,J)) THEN
      ID(J)=1
      GO TO 40
      ENDIF
50    CONTINUE
40    CONTINUE
      DO 45 I=1,3
```

```
        JD(I)=0
        DO 55 J=1,3
        DO 55 K=1,NLOC
        IF (LOC(K).EQ.IEMT(JE,I,J)) THEN
        JD(I)=1
        GO TO 45
        ENDIF
 55     CONTINUE
 45     CONTINUE
        NEX=(NONX-1)/2
        NX=0
        do 70 j=1,nonx
        if (id(1).eq.0.and.j.eq.1) go to 70
        if (id(3).eq.0.and.j.eq.nonx) go to 70
        if (id(2).eq.0.and.mod(j,2).eq.0) go to 70
        NX=NX+1
        XP(NX)=X(j)
 70     CONTINUE
        XP(NX)=X(NONX)                    !TO KEEP SIZE....
        XP(1)=X(1)                        !TO KEEP SIZE....
        NEY=(NONY-1)/2
        NY=0
        do 80 j=1,nony
        if (jd(1).eq.0.and.j.eq.1) go to 80
        if (jd(3).eq.0.and.j.eq.nony) go to 80
        if (jd(2).eq.0.and.mod(j,2).eq.0) go to 80
        NY=NY+1
        YP(ny)=Y(j)
 80     CONTINUE
        YP(NY)=Y(NONY)                 !TO KEEP SIZE....
        YP(1)=Y(1)                     !TO KEEP SIZE....
        NEZ=(NONZ-1)/2
        IF (NNODES.EQ.8) THEN
        NEX=(NONX+1)/2-1
        NEY=(NONY+1)/2-1
        NEZ=(NONZ+1)/2-1
        END IF
        NEXY=NEX*NEY
        NE=NEXY*NEZ
c--     Reading stresses I -el. no.; J-loc. no.
        DO 81 I=1,NE
        DO 81 J=1,NLOC
        READ (IUO,82) (S(I,J,K),K=1,6)
 81     CONTINUE
 82     FORMAT (...,6E15.6)

        NN=(NL...)
        IF (JS.EQ...) NN=NN-1
        NEE=NN*NEXY
c--     stresses of elements nn+1 onwards are required..
        K=0
        DO 89 I=1,NEZ
        N1=NEE+NE...
        N2=N1+NEX...
```

```
      DO 88 JJ=1,3
      IF (JD(JJ).EQ.0) GO TO 88
      DO 90 I=N1,N2
      DO 91 II=1,3
      IF (ID(II).EQ.0) GO TO 91
      IL=IBMT(JS,JJ,II)
      LL=0
      DO 92 L=1,NLOC
   92 IF (IL.EQ.LOC(L)) LL=L
      K=K+1
      ZP(K)=STRES(I,LL,NSI)
      IF (II.NE.1.OR.I.EQ.N1) GO TO 91
      IF (ID(3).EQ.0.OR.ID(1).EQ.0) GO TO 91
      K=K-1
      ZP(K)=(ZP(K)+ZP(K+1))/2.
   91 CONTINUE
   90 CONTINUE
      IJ=K-NX+1
      IF (JJ.NE.1.OR.J.EQ.1) GO TO 88
      IF (JD(3).EQ.0) GO TO 88
      K=K-2*NX
      DO 93 KK=1,NX
      K=K+1
   93 ZP(K)=(ZP(K)+ZP(K+NX))/2.
   88 continue
   99 continue
      do i=1,nv
      jl=1+(i-1)*nx
      ju=jl+nx-1
  226  format (2x,10e11.4)
      end do
      RETURN
      end
```

APPENDIX - D

LISTING OF THE EXAMPLE RESULTS

INPUT DATA for ' PREPROCESSOR' : MAN.INP

```
3 node 21.[02/902]s; delam- mode load: 5x3x5 MESH-man.inp (10/30/8
8                         !ntype of the element (8 OR 21)
5,3.5.0                   ! NO. of coord. in x,y &z dir.. Rad. of hole
0.0, 2.0,4.0,6.0,8.0      !x - coordinates               !Rad. =0 means
0.0, 3.0,6.0              !y - coordinates
0.0, 0.5,1.0,1.5,2.0      !z - coordinates
1. 300.0, 75, 1           !from node no.,nodal temp..to node no., incre
-1, 0.0,  0, 0            !data termination indicator....
1. 300.0, 32, 1           !from el. no., stress free temp., to el. no.,
-1,  0.0,  0, 0           !data termination indicator....
1.  1. 32, 1              !from element no..mat. no..to node no..increm
-1, 0,  0, 0              !data termination indicator....
1, 1, 16, 1               !from el. no..mat.axis orient set no..to node
17, 2, 32, 1              !from el. no..mat.axis orient set no.,to node
-1. 0,  0. 0              !data termination indicator....
1.16                      !from el. no., to el. no., for same stiffness
17.32                     !from el. no., to el. no., for same stiffness
-1.0                      !data termination indicator.. for same stiffness elements
0.2                       !no. of nodes to simulate split, dir. normal to the plane
15                        !no. of double nodes for Delamination Region
3,8.13.18,23,28,33,38,43,48,53.58,63,68,73  !double nodes
0.0,3.0, 0.0,6.0, 1.0,2.0   !x,y.z limits of the solid that has d
EL     1     21.0E06            !Elastic constant...
ET     1     1.7E06             !Elastic constant...
EZ     1     1.7E06             !Elastic constant...
NULT   1     0.3                !Elastic constant... as per elasticity notation
NULZ   1     0.3                !Elastic constant... as per elasticity notation
NUTZ   1     0.54               !Elastic constant...
GLT    1     0.94E06            !Elastic constant...
GLZ    1     0.94E06            !Elastic constant...
GTZ    1     0.50E06            !Elastic constant...
ALFL   1     0.2E-06            !Thermal expansion coefficient in L- d...
ALFT   1     0.16E-04           !Thermal expansion coefficient in T- d...
ALFZ   1     0.16E-04           !Thermal expansion coefficient in Z- d...
-1                              !data termination indicator for Mat. constant
     1     5    25    55     !mat. axis orientation for element set =1
     2    25    75     5     !mat. axis orientation for element set =2
    -1                       !data termination indicator....
    -1                                                   !data termination for force b.c
     4    UY        0.0                  24     5 !from,type,value,to,increment
     5    UY        0.                   25     5 !          of
     1    UX        0.                   51    25 !  disp. boundary conditions
     2    UX        0.                   52    25
     3    UX        0.                   53    25
     4    UX        0.                   54    25
     5    UX        0.                   55    25
     1    UZ        0.0                  71     5
    51    UY      0.006                  75     1 !nonzero disp. b.c
    -1                                            !data termination for disp. b.c
21.0,0.0,0.0,0,0        !stress loc. nos.
```

OUTPUT FROM 'PREPROCESSOR' : [illegible]

1 0 node E1.[02/902]s; delam- [illegible] (10/30/87)

2 90 2 1 0 50

3	10	1	0	1	1	1	1	0.0000	0.0000	0.0000		300.
4	2	1	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
5	3	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
6	4	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
7	5	1	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
8	6	1	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
9	7	0	0	1	1	1	1	2.0000	0.0000	0.0000	0	300.
10	8	0	0	0	1	1	1	2.0000	0.0000	0.5000	0	300.
11	9	0	0	0	1	1	1	2.0000	0.0000	1.0000	0	300.
12	10	0	0	0	1	1	1	2.0000	0.0000	1.0000	0	300.
13	11	0	1	0	1	1	1	2.0000	0.0000	1.5000	0	300.
14	12	0	1	0	1	1	1	2.0000	0.0000	2.0000	0	300.
15	13	0	0	1	1	1	1	4.0000	0.0000	0.0000	0	300.
16	14	0	0	0	1	1	1	4.0000	0.0000	0.5000	0	300.
17	15	0	0	0	1	1	1	4.0000	0.0000	1.0000	0	300.
18	16	0	0	0	1	1	1	4.0000	0.0000	1.0000	0	300.
19	17	0	1	0	1	1	1	4.0000	0.0000	1.5000	0	300.
20	18	0	1	0	1	1	1	4.0000	0.0000	2.0000	0	300.
21	19	0	0	1	1	1	1	6.0000	0.0000	0.0000	0	300.
22	20	0	0	0	1	1	1	6.0000	0.0000	0.5000	0	300.
23	21	0	0	0	1	1	1	6.0000	0.0000	1.0000	0	300.
24	22	0	0	0	1	1	1	6.0000	0.0000	1.0000	0	300.
25	23	0	1	0	1	1	1	6.0000	0.0000	1.5000	0	300.
26	24	0	1	0	1	1	1	6.0000	0.0000	2.0000	0	300.
27	25	0	0	1	1	1	1	8.0000	0.0000	0.0000	0	300.
28	26	0	0	0	1	1	1	8.0000	0.0000	0.5000	0	300.
29	27	0	0	0	1	1	1	8.0000	0.0000	1.0000	0	300.
30	28	0	0	0	1	1	1	8.0000	0.0000	1.0000	0	300.
31	29	0	1	0	1	1	1	8.0000	0.0000	1.5000	0	300.
32	30	0	1	0	1	1	1	8.0000	0.0000	2.0000	0	300.
33	31	1	0	1	1	1	1	0.0000	3.0000	0.0000	0	300.
34	32	1	0	0	1	1	1	0.0000	3.0000	0.5000	0	300.
35	33	1	0	0	1	1	1	0.0000	3.0000	1.0000	0	300.
36	34	1	0	0	1	1	1	0.0000	3.0000	1.0000	0	300.
37	36	1	0	0	1	1	1	0.0000	3.0000	2.0000	1	300.
38	37	0	0	1	1	1	1	2.0000	3.0000	0.0000	0	300.
39	38	0	0	0	1	1	1	2.0000	3.0000	0.5000	0	300.
40	39	0	0	0	1	1	1	2.0000	3.0000	1.0000	0	300.
41	40	0	0	0	1	1	1	2.0000	3.0000	1.0000	0	300.
42	42	0	0	0	1	1	1	2.0000	3.0000	2.0000	1	300.
43	43	0	0	1	1	1	1	4.0000	3.0000	0.0000	0	300.
44	44	0	0	0	1	1	1	4.0000	3.0000	0.5000	0	300.
45	45	0	0	0	1	1	1	4.0000	3.0000	1.0000	0	300.
46	46	0	0	0	1	1	1	4.0000	3.0000	1.0000	0	300.
47	48	0	0	0	1	1	1	4.0000	3.0000	2.0000	1	300.
48	49	0	0	1	1	1	1	6.0000	3.0000	0.0000	0	300.
49	50	0	0	0	1	1	1	6.0000	3.0000	0.5000	0	300.
50	51	0	0	0	1	1	1	6.0000	3.0000	1.0000	0	300.
51	52	0	0	0	1	1	1	6.0000	3.0000	1.0000	0	300.
52	54	0	0	0	1	1	1	6.0000	3.0000	2.0000	1	300.
53	55	0	0	1	1	1	1	6.0000	3.0000	0.0000	0	300.
54	56	0	0	0	1	1	1	6.0000	3.0000	0.5000	0	300.

55	57	0	0	0	1	1	1	8.0000	3.0000	1.0000	0	300.
56	58	0	0	0	1	1	1	8.0000	3.0000	1.0000	0	300.
57	60	0	0	0	1	1	1	8.0000	3.0000	2.0000	1	300.
58	61	1	0	1	1	1	1	0.0000	6.0000	0.0000	0	300.
59	62	1	0	0	1	1	1	0.0000	6.0000	0.5000	0	300.
60	63	1	0	0	1	1	1	0.0000	6.0000	1.0000	0	300.
61	64	1	0	0	1	1	1	0.0000	6.0000	1.0000	0	300.
62	66	1	0	0	1	1	1	0.0000	6.0000	2.0000	1	300.
63	67	0	0	1	1	1	1	2.0000	6.0000	0.0000	0	300.
64	68	0	0	0	1	1	1	2.0000	6.0000	0.5000	0	300.
65	69	0	0	0	1	1	1	2.0000	6.0000	1.0000	0	300.
66	70	0	0	0	1	1	1	2.0000	6.0000	1.0000	0	300.
67	72	0	0	0	1	1	1	2.0000	6.0000	2.0000	1	300.
68	73	0	0	1	1	1	1	4.0000	6.0000	0.0000	0	300.
69	74	0	0	0	1	1	1	4.0000	6.0000	0.5000	0	300.
70	75	0	0	0	1	1	1	4.0000	6.0000	1.0000	0	300.
71	76	0	0	0	1	1	1	4.0000	6.0000	1.0000	0	300.
72	78	0	0	0	1	1	1	4.0000	6.0000	2.0000	1	300.
73	79	0	0	1	1	1	1	6.0000	6.0000	0.0000	0	300.
74	80	0	0	0	1	1	1	6.0000	6.0000	0.5000	0	300.
75	81	0	0	0	1	1	1	6.0000	6.0000	1.0000	0	300.
76	82	0	0	0	1	1	1	6.0000	6.0000	1.0000	0	300.
77	84	0	0	0	1	1	1	6.0000	6.0000	2.0000	1	300.
78	85	0	0	1	1	1	1	8.0000	6.0000	0.0000	0	300.
79	86	0	0	0	1	1	1	8.0000	6.0000	0.5000	0	300.
80	87	0	0	0	1	1	1	8.0000	6.0000	1.0000	0	300.
81	88	0	0	0	1	1	1	8.0000	6.0000	1.0000	0	300.
82	90	0	0	0	1	1	1	8.0000	6.0000	2.0000	1	300.
83	[illegible]	30										
84		1.										
85	61	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
86	62	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
87	63	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
88	65	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
89	66	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
90	67	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
91	68	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
92	69	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
93	71	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
94	72	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
95	73	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
96	74	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
97	75	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
98	77	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
99	78	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
100	79	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
101	80	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
102	81	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
103	83	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
104	84	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
105	85	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
106	86	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
107	87	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		
108	88	1	2	1	7	1	0	0.0060000	0.0000000	0.100E+21		

```
 58    1    2    1    7    [illegible] [illegible]   [illegible] 0.0000000 0.100E+21
 64    1    2    1    7    1    [illegible]   [illegible] 0.0000000 0.100E+21
 70    1    2    1    7    1    [illegible]   [illegible] 0.0000000 0.100E+21
 76    1    2    1    7    1    [illegible]   [illegible] 0.0000000 0.100E+21
 82    1    2    1    7    1    [illegible]   [illegible] 0.0000000 0.100E+21
 88    1    2    1    7    1    [illegible]   [illegible] 0.0000000 0.100E+21
  8   32    1    0    2         [illegible]  .   [illegible]   2
  1    1                     [illegible]-LAYER; [illegible]2==90-LAYER.
     300. 21000000.  1700000.  [illegible]   [illegible]   0.0243    0.5400
  940000.    940000.   500000. [illegible] [illegible] 0.0000160
  1    6   30   66
  2   30   90    6
 21    0    0    0   [illegible] [illegible] [illegible]



         0.

  1                  1    1    1        500.   [illegible]            0
 38   32    2    8   37   [illegible]    1
  2                  1    1    1        [illegible]   [illegible]            1
 44   38    8   14   43   [illegible]   [illegible]   13
  3                  1    1    1        [illegible]   [illegible]            1
 50   44   14   20   49   [illegible]   13   19
  4                  1    1    1        [illegible]   [illegible]            1
 56   50   20   26   55   [illegible]   [illegible]   [illegible]
  5                  1    1    1        [illegible]   [illegible]            1
 68   62   32   38   67   61   31   [illegible]
  6                  1    1    1        [illegible]   [illegible]            1
 74   68   38   44   73   67   37   43
  7                  1    1    1        [illegible]   [illegible]            1
 80   74   44   50   79   [illegible]   43   49
  8                  1    1    1        [illegible]   [illegible]            1
 86   80   50   56   85   [illegible]   49   55
  9                  1    1    1        [illegible]   [illegible]            1
 39   33    3    9   38   [illegible]    2    8
 10                  1    1    1        [illegible]   [illegible]            1
 45   39    9   15   44   [illegible]    8   14
 11                  1    1    1        [illegible]   [illegible]            1
 51   45   15   21   50   44   14   [illegible]
 12                  1    1    1        [illegible]   [illegible]            1
 57   51   21   27   56   50   20   [illegible]
 13                  1    1    1        [illegible]   [illegible]            1
 69   63   33   39   68   [illegible]   [illegible]   [illegible]
 14                  1    1    1        [illegible]   [illegible]            1
 75   69   39   45   74   [illegible]   [illegible]   44
 15                  1    1    1        [illegible]   [illegible]            1
 81   75   45   51   80   [illegible]   44   50
 16                  1    1    1        [illegible]   [illegible]            1
 87   81   51   57   86   [illegible]   50   56
 17                  1    1    1        [illegible]   [illegible]            0
 41   35    5   11   40   34    4   10
 18                  1    1    1        [illegible]   [illegible]            1
 47   41   11   17   46   [illegible]   [illegible]   16
```

```
163   19                     1    2    1         300.    6          1
164   53   47   17   23   52   46   16   22
165   20                     1    2    1         300.   12          1
166   59   53   23   29   58   52   22   28
167   21                     1    2    1         300.    6          1
168   71   65   35   41   70   64   34   40
169   22                     1    2    1         300.    6          1
170   77   71   41   47   76   70   40   46
171   23                     1    2    1         300.    6          1
172   83   77   47   53   82   76   46   52
173   24                     1    2    1         300.    6          1
174   89   83   53   59   88   82   52   58
175   25                     1    2    1         300.    6          1
176   42   36    6   12   41   35    5   11
177   26                     1    2    1         300.    6          1
178   48   42   12   18   47   41   11   17
179   27                     1    2    1         300.    6          1
180   54   48   18   24   53   47   17   23
181   28                     1    2    1         300.   12          1
182   60   54   24   30   59   53   23   29
183   29                     1    2    1         300.    6          1
184   72   66   36   42   71   65   35   41
185   30                     1    2    1         300.    6          1
186   78   72   42   48   77   71   41   47
187   31                     1    2    1         300.    6          1
188   84   78   48   54   83   77   47   53
189   32                     1    2    1         300.    6          1
190   90   84   54   60   89   83   53   59
191
192         1.
193   10
194    3  2  0.0
195    4  2  0.0
196    9  2  0.0
197   10  2  0.0
198   15  2  0.0
199   16  2  0.0
200   21  2  0.0
201   22  2  0.0
202   27  2  0.0
203   28  2  0.0
204   15
205    3    4  1 0 1     1     3
206    9   10  1 0 1     1     8
207   15   16  1 0 1     1    13
208   21   22  1 0 1     1    18
209   27   28  1 0 1     1    23
210   33   34  1 1 1     1    28
211   39   40  1 1 1     1    33
212   45   46  1 1 1     1    38
213   51   52  1 1 1     1    43
214   57   58  1 1 1     1    48
215   63   64  1 1 1     1    53
216   69   70  1 1 1     1    58
```

```
217   75   76  1 1 1      !      63
218   81   82  1 1 1      !      68
219   87   88  1 1 1      !      73
220        0   0   0
221          1   32
222   27   28  1          !        23
223   27   28  2          !        23
224   27   28  3          !        23
225        0   0   0
226          1   32
227   21   22  1          !        18
228   21   22  2          !        18
229   21   22  3          !        18
230   57   58  1          !        48
231   57   58  2          !        48
232   57   58  3          !        48
233        0   0   0
234          1   32
235 9999 9999 0
236
```

MODIFICATIONS FOR CRACK OPENING SEQUENCE

```
   10
    3  2  0.0
    4  2  0.0
    9  2  0.0
   10  2  0.0
   15  2  0.0
   16  2  0.0
   21  2  0.0
   22  2  0.0
   27  2  0.0
   28  2  0.0
   15
    3    4  1 0 1     !       3
    9   10  1 0 1     !       8
   15   16  1 0 1     !      13
   21   22  1 0 1     !      18
   27   28  1 0 1     !      23
   33   34  1 1 1     !      28
   39   40  1 1 1     !      33
   45   46  1 1 1     !      38
   51   52  1 1 1     !      43
   57   58  1 1 1     !      48
   63   64  1 1 1     !      53
   69   70  1 1 1     !      58
   75   76  1 1 1     !      63
   81   82  1 1 1     !      68
   87   88  1 1 1     !      73
        0  0  0
         1  32
   27    28  1          !       23
   27    28  2          !       23
   27    28  3          !       23
        0  0  0
         1  32
   21    22  1          !       18
   21    22  2          !       18
   21    22  3          !       18
   57    58  1          !       48
   57    58  2          !       48
   57    58  3          !       48
        0  0  0
         1  32
 9999 9999 0
```

OUTPUT FROM 'RSAP II : [illegible].DAT

S node E1.C02/5023s; [illegible] (10/30/87)

C O N T R O L   I N F O R M A T I O N

NUMBER OF NODAL POINTS - [illegible]
NUMBER OF ELEMENT TYPES = [illegible]
NUMBER OF LOAD CASES - 1
NUMBER OF FREQUENCIES [illegible]
ANALYSIS CODE (NDYN) - [illegible]
EQ.0, STATIC
EQ.1, MODAL EXTRACTION
EQ.2, FORCED RESPONSE
EQ.3, RESPONSE SPECTRUM
EQ.4, DIRECT INTEGRATION
SOLUTION MODE (MODEX) - [illegible]
EQ.0, EXECUTION
EQ.1, DATA CHECK
NUMBER OF SUBSPACE
ITERATION VECTORS (NAD) - [illegible]
EQUATIONS PER BLOCK - 0
TAPE10 SAVE FLAG (N10SV) - 0

NODAL POINT INPUT DATA

ONODE BOUNDARY CONDITION CODES NODAL POINT COORDINATES
NUMBER X Y Z XX YY ZZ X Y Z

1B O U N D A R Y   E L E M E N T S.

ELEMENT TYPE = [illegible],
NUMBER OF ELEMENTS = [illegible]

ELEMENT LOAD CASE MULTIPLIERS

CASE(A)	CASE(B)	CASE(C)	CASE(D)
1.0000	0.0000	0.0000	0.0000

ELEMENT NUMBER	NODE (N)	NODES DEFINING CONSTRAINT DIRECTION (NI)	(NJ)	(NK)	(NL)	CODE KD	CODE KR	GENERATION CODE (KN)
1	61	1	2	1	7	1	0	0
2	62	1	2	1	7	1	0	0
3	63	1	2	1	7	1	0	0
4	65	1	2	1	7	1	0	0
5	66	1	2	1	7	1	0	0
6	67	1	2	1	7	1	0	0

7	68	1	2	1	7	1	0	0
8	69	1	2	1	7	1	0	0
9	71	1	2	1	7	1	0	0
10	72	1	2	1	7	1	0	0
11	73	1	2	1	7	1	0	0
12	74	1	2	1	7	1	0	0
13	75	1	2	1	7	1	0	0
14	77	1	2	1	7	1	0	0
15	78	1	2	1	7	1	0	0
16	79	1	2	1	7	1	0	0
17	80	1	2	1	7	1	0	0
18	81	1	2	1	7	1	0	0
19	83	1	2	1	7	1	0	0
20	84	1	2	1	7	1	0	0
21	85	1	2	1	7	1	0	0
22	86	1	2	1	7	1	0	0
23	87	1	2	1	7	1	0	0
24	89	1	2	1	7	1	0	0
25	90	1	2	1	7	1	0	0
26	64	1	2	1	7	1	0	0
27	70	1	2	1	7	1	0	0
28	76	1	2	1	7	1	0	0
29	82	1	2	1	7	1	0	0
30	88	1	2	1	7	1	0	0

121 - N O D E S O L I D E L E M E N T I N P U T D A T A

CONTROL INFORMATION

NUMBER OF 21-NODE ELEMENTS = 30

NUMBER OF MATERIAL SETS = 1

MAXIMUM NUMBER OF MATERIAL
TEMPERATURE INPUT POINTS = 1

NUMBER OF MATERIAL
AXIS ORIENTATION SETS = 2

NUMBER OF DISTRIBUTED LOAD SETS = 0

MAXIMUM NUMBER OF ELEMENT NODES = 8

NUMBER OF STRESS OUTPUT SETS = 1

R,S COORDINATE INTEGRATION ORDER = 2

T COORDINATE INTEGRATION ORDER = 2

MATERIAL PROPERTY TABLES

```
MATERIAL NUMBER    = (  1)
 NUMBER OF
 TEMPERATURE POINTS = (  1)
 WEIGHT DENSITY     = ( [illegible])
 MASS  /DENSITY     = ( [illegible])
 IDENTIFICATION     = ([illegible])

 TEMPERATURE         E11          E22          E33      V12    V13    V23        G12

      500.00  21000000.0  [illegible]  [illegible]  0.024  0.024  0.540   940000.0


M A T E R I A L   A X I S   O R I E N T A T I O N   T A B L E

    SET    NODE    NODE    NODE
 NUMBER      NI      NJ      NK

      1       6      30     [illegible]
      2      30      90     [illegible]


S T R E S S   O U T P U T   R E Q U E S T   T A B L E

    SET   POINT   POINT   POINT   POINT   POINT   POINT   POINT
 NUMBER     (1)     (2)     (3)     (4)     (5)     (6)     (7)

      1      21       0       0       0       0       0       0



E L E M E N T   L O A D   C A S E   M U L T I P L I E R S

                                   CASE A    CASE B    CASE C    CASE D

 X-DIRECTION GRAVITY =               0.00      0.00      0.00      0.00
 Y-DIRECTION GRAVITY =               0.00      0.00      0.00      0.00
 Z-DIRECTION GRAVITY =               0.00      0.00      0.00      0.00
 THERMAL LOADING     =               0.00      0.00      0.00      0.00
 PRESSURE LOADING    =               0.00      0.00      0.00      0.00


3 / D   8   T O   2 1   N O D E   S O L I D   E L E M E N T   D A T A

 ELEMENT  NODES  NODES  MATL.  MATL.  STRESS   STRESS  NODE  GAUSS  GAUSS
  NUMBER -NDIS- -NXYZ-  TABLE   AXIS  OUTPUT    FREE   INC.   PTS.   PTS.  MATR
                          NO.    SET     SET   TEMP.  -KC-  -R,S-   -T-  RE-U
       1      8      8      1      1       1   500.0     1      2      0

       2      8      8      1      1       1   500.0     6      2      0

       3      8      8      1      1       1   500.0     6      2      0

       4      8      8      1      1       1   500.0    12      2      0

```

5	8	8	1	1	1	500.0	6	2	0
6	8	8	1	1	1	500.0	6	2	0
7	8	8	1	1	1	500.0	6	2	0
8	8	8	1	1	1	500.0	1	2	0
9	8	8	1	1	1	500.0	6	2	0
10	8	8	1	1	1	500.0	6	2	0
11	8	8	1	1	1	500.0	6	2	0
12	8	8	1	1	1	500.0	12	2	0
13	8	8	1	1	1	500.0	6	2	0
14	8	8	1	1	1	500.0	6	2	0
15	8	8	1	1	1	500.0	6	2	0
16	8	8	1	1	1	500.0	1	2	0
17	8	8	1	2	1	500.0	1	2	0
18	8	8	1	2	1	500.0	6	2	0
19	8	8	1	2	1	500.0	6	2	0
20	8	8	1	2	1	500.0	12	2	0
21	8	8	1	2	1	500.0	6	2	0
22	8	8	1	2	1	500.0	6	2	0
23	8	8	1	2	1	500.0	6	2	0
24	8	8	1	2	1	500.0	1	2	0
25	8	8	1	2	1	500.0	6	2	0
26	8	8	1	2	1	500.0	6	2	0
27	8	8	1	2	1	500.0	6	2	0
28	8	8	1	2	1	500.0	12	2	0
29	8	8	1	2	1	500.0	6	2	0
30	8	8	1	2	1	500.0	6	2	0
31	8	8	1	2	1	500.0	6	2	0

32 6 6 . . . 300.0 1 2 0

EQUATION PARAMETERS

TOTAL NUMBER OF EQUATIONS [illegible]
BANDWIDTH [illegible]
NUMBER OF EQUATIONS IN A BLOCK [illegible]
NUMBER OF BLOCKS [illegible]

NODAL LOADS (STATIC) OR MASSES (DYNAMIC)

NODE NUMBER	LOAD CASE	X-AXIS FORCE	Y-AXIS FORCE	Z-AXIS FORCE	X-AXIS MOMENT

STRUCTURE LOAD CASE	ELEMENT LOAD MULTIPLIERS A	B	C	D
1	1.000	0.000	0.000	0.000

3	2	0.000000000000000E+00
4	2	0.000000000000000E+00
9	2	0.000000000000000E+00
10	2	0.000000000000000E+00
15	2	0.000000000000000E+00
16	2	0.000000000000000E+00
21	2	0.000000000000000E+00
22	2	0.000000000000000E+00
27	2	0.000000000000000E+00
28	2	0.000000000000000E+00

npair = 15

3	4	1	0	1
9	10	1	0	1
15	16	1	0	1
21	22	1	0	1
27	28	1	0	1
33	34	1	1	1
39	40	1	1	1
45	46	1	1	1
51	52	1	1	1
57	58	1	1	1
63	64	1	1	1
69	70	1	1	1
75	76	1	1	1
81	82	1	1	1
87	88	1	1	1

--- NODES --- DEGREES RELEASED ---
0 0 0

***** STEP # 0 *******

^^

---- NODAL DISPLACEMENTS AND FORCES IN SOLEU---

****

NODE	U	V	W	Fx	Fy	Fz
1	0.00000E+00	0.11463E-02	0.00000E+00	0.00000E+00	0.54179E-13	0.00000E+00
2	0.00000E+00	0.88914E-03	-0.23028E-03	0.00000E+00	0.31974E-13	-0.21316E-13
3	0.00000E+00	0.00000E+00	-0.48447E-03	0.00000E+00	-1254.5	-12.449
4	0.00000E+00	0.00000E+00	-0.48447E-03	0.00000E+00	-5360.6	12.449
5	0.00000E+00	0.00000E+00	-0.70911E-03	0.00000E+00	0.00000E+00	-0.99476E-13
6	0.00000E+00	0.00000E+00	-0.93084E-03	0.00000E+00	0.00000E+00	-0.99476E-13
7	-0.73276E-04	0.11457E-02	0.00000E+00	-0.13434E-13	0.25802E-12	0.00000E+00
8	-0.75126E-04	0.88872E-03	-0.23073E-03	0.53291E-13	-0.92371E-13	-0.92371E-13
9	-0.78116E-04	0.00000E+00	-0.48515E-03	-14.750	-2505.8	-27.415
10	-0.78116E-04	0.00000E+00	-0.48515E-03	14.750	-10721.	27.415
11	-0.80760E-04	0.00000E+00	-0.70685E-03	0.49738E-13	0.00000E+00	-0.56843E-13
12	-0.82216E-04	0.00000E+00	-0.92999E-03	-0.71054E-14	0.00000E+00	-0.85265E-13
13	-0.13991E-03	0.11383E-02	0.00000E+00	-0.29310E-13	0.30509E-12	0.00000E+00
14	-0.14765E-03	0.88319E-03	-0.23491E-03	0.88818E-13	-0.87041E-13	-0.37659E-12
15	-0.15902E-03	0.00000E+00	-0.49065E-03	-74.970	-2492.4	-64.931
16	-0.15902E-03	0.00000E+00	-0.49065E-03	74.970	-10718.	64.931
17	-0.17076E-03	0.00000E+00	-0.71501E-03	-0.71054E-14	0.00000E+00	-0.14211E-13
18	-0.16835E-03	0.00000E+00	-0.93503E-03	0.71054E-14	0.00000E+00	-0.54001E-12
19	-0.18363E-03	0.11366E-02	0.00000E+00	0.10658E-13	0.78160E-13	0.00000E+00
20	-0.20636E-03	0.88057E-03	-0.25799E-03	-0.65725E-13	0.18296E-12	0.00000E+00
21	-0.25376E-03	0.00000E+00	-0.53085E-03	-224.03	-2509.2	-90.916
22	-0.25376E-03	0.00000E+00	-0.53085E-03	224.03	-10719.	90.916
23	-0.30927E-03	0.00000E+00	-0.74951E-03	-0.10658E-12	0.00000E+00	0.55422E-12
24	-0.35406E-03	0.00000E+00	-0.94473E-03	-0.39080E-13	0.00000E+00	-0.22737E-12
25	-0.20854E-03	0.12250E-02	0.00000E+00	0.16376E-13	-0.55067E-13	0.00000E+00
26	-0.23900E-03	0.94963E-03	-0.17825E-03	0.35527E-14	0.81712E-13	-0.28422E-13
27	-0.37387E-03	0.00000E+00	-0.39000E-03	-389.81	-1305.8	104.43
28	-0.37387E-03	0.00000E+00	-0.39000E-03	389.81	-5368.9	-104.43
29	-0.55838E-03	0.00000E+00	-0.57126E-03	-0.97700E-13	0.00000E+00	0.22737E-12
30	-0.77311E-03	0.00000E+00	-0.74140E-03	-0.46185E-13	0.00000E+00	-0.14211E-12
31	0.00000E+00	0.30734E-02	0.00000E+00	0.00000E+00	0.23093E-13	0.00000E+00
32	0.00000E+00	0.30562E-02	-0.18195E-03	0.00000E+00	0.17941E-12	-0.16342E-12
33	0.00000E+00	0.30773E-02	-0.41680E-03	0.00000E+00	-444.89	-53.583
34	0.00000E+00	0.30773E-02	-0.41680E-03	0.00000E+00	444.89	53.583
35	0.00000E+00	0.30191E-02	-0.64568E-03	0.00000E+00	0.17935E-11	0.11369E-12
36	0.00000E+00	0.30104E-02	-0.86908E-03	0.00000E+00	-0.11990E-12	-0.24158E-12
37	-0.84036E-04	0.30730E-02	0.00000E+00	0.71054E-14	0.41656E-12	0.00000E+00
38	-0.85401E-04	0.30559E-02	-0.18228E-03	0.74607E-13	-0.52935E-12	-0.41922E-12
39	-0.87090E-04	0.30771E-02	-0.41734E-03	-19.334	-888.27	-112.18
40	-0.87090E-04	0.30771E-02	-0.41734E-03	19.334	888.27	112.18
41	-0.87638E-04	0.30190E-02	-0.64533E-03	-0.88818E-13	0.18510E-11	0.17053E-12
42	-0.88506E-04	0.30104E-02	-0.86809E-03	-0.71054E-14	0.11431E-11	-0.52580E-12
43	-0.16394E-03	0.30731E-02	0.00000E+00	-0.35527E-13	0.29488E-12	0.00000E+00
44	-0.17014E-03	0.30560E-02	-0.18640E-03	0.11724E-12	-0.31442E-12	-0.83844E-12
45	-0.17832E-03	0.30769E-02	-0.42251E-03	-125.37	-885.34	-186.00
46	-0.17832E-03	0.30769E-02	-0.42251E-03	125.37	885.34	186.00
47	-0.16532E-03	0.30190E-02	-0.65064E-03	0.31974E-13	-0.91357E-12	0.15632E-12

```
48-0.18212E-03 0.30[illegible]E-02-0.[illegible]E-03 0.[illegible]E-13 0.22280E-11-0.63949E-12
49-0.22847E-03 0.307[illegible]E-02 0.00000E+00-0.15721E-12 0.98588E-13 0.00000E+00
50-0.24583E-03 0.30[illegible]E-02-0.[illegible]E-03 0.20466E-12-0.43165E-12-0.20606E-12
51-0.28415E-03 0.30775E-02-0.4[illegible]E-03 -408.01     -900.19     -243.72
52-0.28415E-03 0.30775E-02-0.4[illegible]E-03  408.01      900.19      243.72
53-0.33013E-03 0.30190E-02-0.[illegible]E-03 0.14921E-12 0.13074E-11 0.48317E-12
54-0.37055E-03 0.30099E-02-0.[illegible]E-03-0.14211E-12 0.15103E-11-0.44054E-12
55-0.26533E-03 0.30814E-02 0.00000E+00-0.10[illegible]E-12-0.88818E-15 0.00000E+00
56-0.30213E-03 0.30[illegible]E-02-0.13[illegible]E-03-0.71054E-14 0.12923E-12 0.48850E-13
57-0.40790E-03 0.30[illegible]E-02-0.[illegible]E-03 -741.90     -469.95      173.82
58-0.40790E-03 0.30[illegible]E-02-0.[illegible]E-03  741.90      469.95     -173.82
59-0.67171E-03 0.30[illegible]E-02-0.[illegible]E-03-0.70166E-13 0.15967E-11 0.14744E-12
60-0.78089E-03 0.301[illegible]E-02-0.[illegible]E-03-0.12[illegible]E-12 0.25624E-12-0.16375E-12
61 0.00000E+00 0.60000E-02 0.00000E+00 0.00000E+00 -16.000      0.00000E+00
62 0.00000E+00 0.60000E-02-0.[illegible]E-03 0.00000E+00  32.000     -0.85265E-13
63 0.00000E+00 0.60000E-02-0.541[illegible]E-03 0.00000E+00 -2416.0      63.375
64 0.00000E+00 0.60000E-02-0.541[illegible]E-03 0.00000E+00  2416.0     -63.375
65 0.00000E+00 0.60000E-02-0.[illegible]E-03 0.00000E+00 -16.000      0.39790E-12
66 0.00000E+00 0.60000E-02-0.[illegible]E-03 0.00000E+00  32.000     -0.56843E-13
67-0.88973E-04 0.60000E-02 0.00000E+00-0.14211E-13  32.000      0.00000E+00
68-0.90275E-04 0.60000E-02-0.[illegible]E-03-0.10658E-13  80.000     -0.29843E-12
69-0.91885E-04 0.60000E-02-0.542[illegible]E-03 -7.6169     -4848.0      124.28
70-0.91885E-04 0.60000E-02-0.542[illegible]E-03  7.6169      4848.0     -124.28
71-0.92281E-04 0.60000E-02-0.7[illegible]E-03-0.39080E-13  48.000     -0.71054E-13
72-0.93165E-04 0.60000E-02-0.9[illegible]E-03-0.42633E-13  16.000     -0.56843E-12
73-0.17348E-03 0.60000E-02 0.00000E+00-0.[illegible]E-13 0.00000E+00 0.00000E+00
74-0.17961E-03 0.60000E-02-0.[illegible]E-03-0.30198E-13  32.000     -0.32552E-12
75-0.18760E-03 0.60000E-02-0.54[illegible]E-03 -58.135     -4928.0      87.420
76-0.18760E-03 0.60000E-02-0.54[illegible]E-03  58.135      4816.0     -87.420
77-0.19438E-03 0.60000E-02-0.7[illegible]E-03-0.12479E-12  32.000      0.10747E-12
78-0.19082E-03 0.60000E-02-0.[illegible]E-03 0.10214E-13 -32.000      0.13189E-12
79-0.24146E-03 0.60000E-02 0.00000E+00 0.14388E-12  32.000      0.00000E+00
80-0.25882E-03 0.60000E-02-0.[illegible]E-03 0.60396E-13  32.000     -0.15810E-12
81-0.29749E-03 0.60000E-02-0.[illegible]E-03 -200.15     -4848.0      58.284
82-0.29749E-03 0.60000E-02 0.[illegible]E-03  200.15      4848.0     -58.284
83-0.34226E-03 0.60000E-02-0.77[illegible]E-03 0.40850E-13  16.000     -0.13589E-12
84-0.38280E-03 0.60000E-02-0.9[illegible]E-03-0.55955E-13  48.000      0.41478E-12
85-0.30106E-03 0.60000E-02 0.00000E+00 0.10658E-13  32.000      0.0000CE+00
86-0.31903E-03 0.60000E-02-0.24017E-03-0.[illegible]E-13  48.000     -0.14211E-13
87-0.42430E-03 0.60000E-02-0.45[illegible]E-03 -374.35     -2432.0      179.58
88-0.42430E-03 0.60000E-02-0.45[illegible]E-03  374.35      2432.0     -179.58
89-0.69559E-03 0.60000E-02-0.[illegible]E-03 0.56843E-13 -16.000      0.88818E-13
90-0.80112E-03 0.60000E-02-0.7[illegible]E-03-0.60396E-13 -32.000      0.12079E-12


--------------------------------------------------------------------------------
          27          28          1
          27          28          2
          27          28          3
           0           0          0
1 $$$$$     STEP #     1  $$$$$$$
----------------------------------------------
```

****

---- NODAL DISPLACEMENTS AND FORCES IN SOLEG---

****

NODE	U	V	W	Fx	Fy	Fz
1	0.00000E+00	0.11427E-02	0.00000E+00	0.00000E+00	0.13145E-12	0.00000E+00
2	0.00000E+00	0.88664E-03	-0.23276E-03	0.00000E+00	-0.71054E-13	-0.49738E-13
3	0.00000E+00	0.00000E+00	-0.48779E-03	0.00000E+00	-1248.7	-12.640
4	0.00000E+00	0.00000E+00	-0.48779E-03	0.00000E+00	-5360.1	12.640
5	0.00000E+00	0.00000E+00	-0.71327E-03	0.00000E+00	0.00000E+00	-0.14211E-12
6	0.00000E+00	0.00000E+00	-0.93787E-03	0.00000E+00	0.00000E+00	-0.14211E-12
7	-0.51868E-04	0.11452E-02	0.00000E+00	-0.11546E-13	0.29132E-12	0.00000E+00
8	-0.50283E-04	0.88797E-03	-0.22883E-03	-0.14211E-13	-0.94147E-13	-0.13500E-12
9	-0.51137E-04	0.00000E+00	-0.48819E-03	17.087	-2504.5	2.0581
10	-0.51137E-04	0.00000E+00	-0.48819E-03	-17.087	-10707.	-2.0581
11	-0.59647E-04	0.00000E+00	-0.71955E-03	-0.42633E-13	0.00000E+00	-0.32685E-12
12	-0.64550E-04	0.00000E+00	-0.94242E-03	-0.39080E-13	0.00000E+00	0.14211E-13
13	-0.99390E-04	0.11508E-02	0.00000E+00	-0.35527E-14	0.14655E-12	0.00000E+00
14	-0.10816E-03	0.89353E-03	-0.21788E-03	0.35265E-13	-0.16165E-12	-0.36238E-12
15	-0.93351E-04	0.00000E+00	-0.44304E-03	26.245	-2539.3	36.102
16	-0.93351E-04	0.00000E+00	-0.44304E-03	-26.245	-10727.	-36.102
17	-0.61080E-04	0.00000E+00	-0.65871E-03	-0.35527E-13	0.00000E+00	-0.85265E-12
18	-0.63022E-04	0.00000E+00	-0.90913E-03	-0.21316E-13	0.00000E+00	0.42633E-13
19	-0.12074E-03	0.11892E-02	0.00000E+00	0.33307E-13	-0.93259E-14	0.00000E+00
20	-0.12896E-03	0.91702E-03	-0.34279E-03	-0.66613E-13	0.77272E-13	0.42633E-13
21	-0.20197E-03	0.00000E+00	-0.71786E-03	-589.53	-2567.7	-622.07
22	-0.20197E-03	0.00000E+00	-0.71786E-03	589.53	-8701.9	622.07
23	0.42192E-04	0.00000E+00	-0.98827E-03	0.35527E-13	0.00000E+00	-0.36948E-12
24	0.44164E-03	0.00000E+00	-0.12639E-02	0.14566E-12	0.00000E+00	-0.41211E-12
25	-0.12269E-03	0.14199E-02	0.00000E+00	0.24425E-13	-0.93259E-14	0.00000E+00
26	-0.15222E-03	0.11143E-02	-0.22936E-03	-0.54623E-13	0.58620E-13	0.42633E-13
27	-0.17158E-03	0.00000E+00	-0.51504E-03	0.71054E-13	-1369.0	-0.85265E-13
28	-0.14282E-02	0.34805E-02	-0.50737E-02	0.12434E-13	0.24158E-12	0.82423E-12
29	-0.38165E-03	0.00000E+00	-0.49310E-02	-0.15632E-12	0.00000E+00	-0.31264E-12
30	0.52385E-03	0.00000E+00	-0.50311E-02	0.00000E+00	0.00000E+00	-0.28422E-12
31	0.00000E+00	0.30678E-02	0.00000E+00	0.00000E+00	0.38192E-13	0.00000E+00
32	0.00000E+00	0.30513E-02	-0.18176E-03	0.00000E+00	-0.27001E-12	-0.19185E-12
33	0.00000E+00	0.30742E-02	-0.41647E-03	0.00000E+00	-433.15	-54.412
34	0.00000E+00	0.30742E-02	-0.41647E-03	0.00000E+00	433.15	54.412
35	0.00000E+00	0.30174E-02	-0.64499E-03	0.00000E+00	0.80291E-12	-0.56343E-13
36	0.00000E+00	0.30086E-02	-0.86801E-03	0.00000E+00	0.79758E-12	-0.24158E-12
37	-0.89885E-04	0.30625E-02	0.00000E+00	0.39080E-13	0.21139E-12	0.00000E+00
38	-0.90917E-04	0.30476E-02	-0.18077E-03	0.24869E-13	0.47606E-12	-0.52580E-12
39	-0.90673E-04	0.30760E-02	-0.41466E-03	18.308	-869.22	-93.926
40	-0.90673E-04	0.30760E-02	-0.41466E-03	-18.308	869.22	93.926
41	-0.87783E-04	0.30188E-02	-0.64290E-03	-0.11724E-12	0.30838E-11	-0.92371E-12
42	-0.87973E-04	0.30083E-02	-0.86669E-03	-0.49738E-13	0.39568E-12	-0.18474E-12
43	-0.17498E-03	0.30748E-02	0.00000E+00	0.33489E-13	-0.26645E-12	0.00000E+00
44	-0.18387E-03	0.30516E-02	-0.18576E-03	0.31619E-12	0.13679E-12	-0.33134E-12
45	-0.18970E-03	0.30600E-02	-0.41987E-03	-32.290	-960.13	-121.99
46	-0.18970E-03	0.30600E-02	-0.41987E-03	32.290	960.13	121.99

```
47-0.18275E-03 0.3[illegible]E-02-[illegible]E-03-[illegible]E-13 0.93792E-12-0.63949E-12
48-0.17762E-03 0.30051E-02-0.[illegible]E-03 0.11369E-12 0.23170E-11 0.36948E-12
49-0.24254E-03 0.321[illegible]E-02 0.00000E+00-0.13323E-13-0.13145E-12 0.00000E+00
50-0.26493E-03 0.31994E-02-0.[illegible]E-03-0.23981E-12 0.40945E-12-0.99476E-13
51-0.32906E-03 0.31915E-02-0.44[illegible]E-03 -[illegible].68      -808.58      -347.65
52-0.32906E-03 0.31915E-02-0.44[illegible]E-03  [illegible].68       808.58       347.65
53-0.36051E-03 0.30615E-02-0.[illegible]E-03-0.19540E-13 0.20961E-12 0.24158E-12
54-0.37399E-03 0.29663E-02-0.[illegible]E-03 0.[illegible]45E-14 0.18332E-11-0.36948E-12
55-0.30339E-03 0.33490E-02 0.00000E+00-0.1[illegible]78E-12 0.34417E-13 0.00000E+00
56-0.32779E-03 0.3377[illegible]E-02-0.[illegible]E-04-0.19[illegible]E-12-0.65281E-13 0.18874E-12
57-0.46466E-03 0.36043E-02-0.[illegible]E-03 -[illegible].01      -79.197       368.55
58-0.46466E-03 0.36043E-02-0.[illegible]E-03  [illegible].01       79.197      -368.55
59-0.67718E-03 0.35075E-02-0.41[illegible]E-03-0.2[illegible]5E-12-0.12574E-11-0.23093E-12
60-0.72348E-03 0.28894E-02-0.[illegible]E-03-0.421[illegible]E-14 0.13501E-11-0.50981E-12
61 0.00000E+00 0.60000E-02 0.00000E+00 0.00000E+00 -16.000       0.00000E+00
62 0.00000E+00 0.60000E-02-0.[illegible]E-03 0.00000E+00  32.000      -0.12790E-12
63 0.00000E+00 0.60000E-02-0.53944E-03 0.00000E+00 -2416.0        62.806
64 0.00000E+00 0.60000E-02-0.53944E-03 0.00000E+00  2416.0       -62.806
65 0.00000E+00 0.60000E-02-0.74043E-03 0.00000E+00 -16.000        0.85265E-13
66 0.00000E+00 0.60000E-02-0.95447E-03 0.00000E+00  32.000        0.11369E-12
67-0.10239E-03 0.60000E-02 0.00000E+00 0.71054E-14  32.000        0.00000E+00
68-0.10480E-03 0.60000E-02-0.2[illegible]21E-03-0.63949E-13  48.000       -0.34106E-12
69-0.10772E-03 0.60000E-02-0.54[illegible]E-03 -4.8732      -4864.0        121.46
70-0.10772E-03 0.60000E-02-0.54[illegible]E-03  4.8732       4896.0       -121.46
71-0.10817E-03 0.60000E-02-0.74[illegible]E-03 0.[illegible]E-13  16.000       -0.12221E-11
72-0.10861E-03 0.60000E-02-0.9[illegible]E-03-0.42633E-13  48.000        0.14211E-12
73-0.19891E-03 0.60000E-02 0.00000E+00 0.31974E-13 -32.000        0.00000E+00
74-0.20703E-03 0.60000E-02-0.26553E-03 0.51514E-13  64.000       -0.24514E-12
75-0.22297E-03 0.60000E-02-0.54440E-03 -59.615      -4928.0        100.92
76-0.22297E-03 0.60000E-02-0.54440E-03  59.615       4832.0       -100.92
77-0.24100E-03 0.60000E-02-0.745[illegible]E-03-0.47513E-13  64.000       -0.82645E-12
78-0.24622E-03 0.60000E-02-0.954[illegible]E-03 0.73275E-13 -16.000        0.41256E-12
79-0.27460E-03 0.60000E-02 0.00000E+00 0.55067E-13 -16.000        0.00000E+00
80-0.29576E-03 0.60000E-02-0.27[illegible]E-03-0.13145E-12  64.000       -0.77272E-13
81-0.34303E-03 0.60000E-02-0.522[illegible]E-03 -221.86      -4640.0        206.30
82-0.34303E-03 0.60000E-02-0.522[illegible]E-03  221.86       4688.0       -206.30
83-0.42417E-03 0.60000E-02-0.70[illegible]E-03 0.1[illegible]99E-13  80.000        0.33573E-12
84-0.50061E-03 0.60000E-02-0.67751E-03-0.41744E-13  16.000        0.22826E-12
85-0.33558E-03 0.60000E-02 0.00000E+00 0.63949E-13  16.000        0.00000E+00
86-0.35735E-03 0.60000E-02-0.[illegible]E-03-0.10[illegible]E-12 0.00000E+00-0.29310E-13
87-0.47481E-03 0.60000E-02-0.[illegible]E-03 -[illegible].50      -2096.0        354.50
88-0.47481E-03 0.60000E-02-0.[illegible]E-03  [illegible].50       2112.0       -354.50
89-0.81700E-03 0.60000E-02-0.[illegible]E-03-0.2131[illegible]E-13 0.00000E+00 0.10658E-12
90-0.10120E-02 0.60000E-02-0.4[illegible]E-03-0.40[illegible]E-13  16.000        0.26645E-13


------------------------  ---------------------------------------------------
        21          22          1
        21          22          2
        21          22          3
        57          58          1
        57          58          2
        57          58          3
         0           0          0
```

1 $$$$$ STEP # 2 $$$$$$$

****

---- NODAL DISPLACEMENTS AND FORCES IN SOLID---

****

NODE	U	V	W	Fx	Fy	Fz
1	0.00000E+00	0.11479E-02	0.00000E+00	0.00000E+00	0.47519E-13	0.00000E+00
2	0.00000E+00	0.88565E-03	-0.22519E-03	0.00000E+00	-0.17764E-12	-0.11369E-12
3	0.00000E+00	0.00000E+00	-0.48940E-03	0.00000E+00	-1257.5	19.659
4	0.00000E+00	0.00000E+00	-0.48940E-03	0.00000E+00	-5347.2	-19.659
5	0.00000E+00	0.00000E+00	-0.72839E-03	0.00000E+00	0.00000E+00	-0.39476E-13
6	0.00000E+00	0.00000E+00	-0.95189E-03	0.00000E+00	0.00000E+00	-0.11369E-12
7	-0.43660E-04	0.11561E-02	0.00000E+00	0.14211E-13	0.11102E-12	0.00000E+00
8	-0.51390E-04	0.89754E-03	-0.21430E-03	0.35527E-14	0.22560E-12	-0.16342E-12
9	-0.26748E-04	0.00000E+00	-0.43272E-03	95.531	-2546.6	77.612
10	-0.26748E-04	0.00000E+00	-0.43272E-03	-95.531	-10733.	-77.612
11	0.24181E-04	0.00000E+00	-0.64642E-03	-0.49738E-13	0.00000E+00	0.85265E-13
12	0.17142E-04	0.00000E+00	-0.90242E-03	-0.10658E-13	0.00000E+00	-0.56843E-13
13	-0.66606E-04	0.11869E-02	0.00000E+00	-0.91432E-13	0.58620E-13	0.00000E+00
14	-0.66621E-04	0.91686E-03	-0.34559E-03	0.00000E+00	0.17764E-14	-0.19185E-12
15	-0.14033E-03	0.00000E+00	-0.72966E-03	-606.80	-2564.5	-666.75
16	-0.14033E-03	0.00000E+00	-0.72966E-03	606.80	-9013.7	666.75
17	0.17186E-03	0.00000E+00	-0.10019E-02	-0.17408E-12	0.00000E+00	-0.48317E-12
18	0.67653E-03	0.00000E+00	-0.12980E-02	-0.13500E-12	0.00000E+00	0.21316E-12
19	-0.84855E-04	0.13739E-02	0.00000E+00	0.39080E-13	0.36415E-12	0.00000E+00
20	-0.96202E-04	0.10742E-02	-0.23259E-03	-0.17764E-13	-0.22915E-12	0.24158E-12
21	-0.70549E-04	0.00000E+00	-0.51319E-03	-0.29310E-13	-2778.4	-0.10658E-12
22	-0.13701E-02	0.32522E-02	-0.57535E-02	0.14211E-13	0.31974E-12	-0.10232E-11
23	-0.22838E-03	0.00000E+00	-0.56503E-02	-0.55422E-12	0.00000E+00	-0.26432E-11
24	0.59570E-03	0.00000E+00	-0.57034E-02	-0.71054E-14	0.00000E+00	0.36380E-11
25	-0.11507E-03	0.14984E-02	0.00000E+00	0.29921E-13	0.47962E-13	0.00000E+00
26	-0.12385E-03	0.11694E-02	-0.27357E-03	-0.39706E-13	0.37303E-13	0.11369E-12
27	-0.15067E-03	0.00000E+00	-0.60236E-03	0.10125E-12	-1498.4	-0.71054E-13
28	-0.16199E-02	0.28741E-02	-0.88858E-02	0.67502E-13	0.29843E-12	0.68212E-12
29	-0.10713E-02	0.00000E+00	-0.89213E-02	0.14211E-13	0.00000E+00	-0.41496E-11
30	-0.29968E-03	0.00000E+00	-0.89224E-02	0.39790E-12	0.00000E+00	0.30127E-11
31	0.00000E+00	0.30553E-02	0.00000E+00	0.00000E+00	0.26823E-12	0.00000E+00
32	0.00000E+00	0.30423E-02	-0.17981E-03	0.00000E+00	-0.17941E-12	-0.49027E-12
33	0.00000E+00	0.30750E-02	-0.41112E-03	0.00000E+00	-433.39	-31.994
34	0.00000E+00	0.30750E-02	-0.41112E-03	0.00000E+00	433.39	31.994
35	0.00000E+00	0.30189E-02	-0.64004E-03	0.00000E+00	0.21174E-11	0.42633E-13
36	0.00000E+00	0.30085E-02	-0.86547E-03	0.00000E+00	0.21272E-12	-0.32685E-12
37	-0.93072E-04	0.30743E-02	0.00000E+00	0.39080E-13	-0.24336E-12	0.00000E+00
38	-0.97256E-04	0.30510E-02	-0.18087E-03	0.10481E-12	0.44764E-12	-0.67502E-12
39	-0.96425E-04	0.30616E-02	-0.41379E-03	79.605	-958.20	-46.563
40	-0.96425E-04	0.30616E-02	-0.41379E-03	-79.605	958.20	46.563
41	-0.83153E-04	0.30059E-02	-0.63399E-03	-0.85265E-13	0.14992E-11	-0.18474E-12
42	-0.77701E-04	0.30000E-02	-0.86053E-03	-0.15987E-13	0.13709E-11	-0.46896E-12

```
43-0.17673E-03 0.32280E-02 0.00000E+00-0.15721E-12-0.25580E-12 0.00000E+00
44-0.19037E-03 0.32062E-02-0.15822E-03 0.28626E-13 0.59686E-12-0.41922E-12
45-0.22271E-03 0.31866E-02-0.42266E-03 -579.02      -819.24      -408.11
46-0.22271E-03 0.31866E-02-0.42266E-03  579.02       819.24       408.11
47-0.20486E-03 0.30612E-02-0.64684E-03-0.10303E-12-0.44942E-12-0.59686E-12
48-0.18848E-03 0.29922E-02-0.86920E-03 0.35527E-14 0.32807E-11 0.15632E-12
49-0.24670E-03 0.33942E-02 0.00000E+00-0.78160E-13 0.44231E-12 0.00000E+00
50-0.25784E-03 0.34314E-02-0.15202E-03 0.20961E-12-0.87574E-12 0.60396E-13
51-0.35287E-03 0.36820E-02-0.43067E-03 -1021.9       72.417      -185.66
52-0.35287E-03 0.36820E-02-0.43067E-03  1021.9      -72.417       185.66
53-0.38131E-03 0.34879E-02-0.65166E-03-0.39968E-12 0.18048E-11-0.14140E-11
54-0.22594E-03 0.26038E-02-0.94266E-03-0.16829E-12-0.11637E-11 0.16129E-11
55-0.29875E-03 0.35344E-02 0.00000E+00-0.24869E-13 0.12068E-12 0.00000E+00
56-0.32107E-03 0.35647E-02-0.15633E-03-0.15099E-12-0.56621E-14 0.25846E-12
57-0.31717E-03 0.37435E-02-0.56635E-03-0.51514E-13 0.19318E-13-0.45119E-12
58-0.11466E-02 0.39825E-02-0.17455E-02-0.35527E-13 0.10107E-11 0.44764E-12
59-0.71109E-03 0.34814E-02-0.19112E-02 0.35527E-13 0.20410E-11-0.30198E-11
60-0.38445E-03 0.25762E-02-0.21761E-02 0.25663E-12-0.37104E-12 0.13705E-11
61 0.00000E+00 0.60000E-02 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
62 0.00000E+00 0.60000E-02-0.28672E-03 0.00000E+00  48.000     -0.21316E-12
63 0.00000E+00 0.60000E-02-0.54480E-03 0.00000E+00 -2448.0       60.814
64 0.00000E+00 0.60000E-02-0.54480E-03 0.00000E+00  2432.0      -60.814
65 0.00000E+00 0.60000E-02-0.74325E-03 0.00000E+00 -32.000      0.22737E-12
66 0.00000E+00 0.60000E-02-0.95227E-03 0.00000E+00  32.000     -0.17053E-12
67-0.10561E-03 0.60000E-02 0.00000E+00 0.71054E-14  16.000      0.00000E+00
68-0.10613E-03 0.60000E-02-0.28179E-03 0.28422E-13  48.000     -0.30553E-12
69-0.11717E-03 0.60000E-02-0.53992E-03 -1.1518      -4896.0       142.11
70-0.11717E-03 0.60000E-02-0.53992E-03  1.1518       4848.0      -142.11
71-0.13113E-03 0.60000E-02-0.74081E-03-0.67502E-13  32.000     -0.42633E-13
72-0.14211E-03 0.60000E-02-0.95069E-03-0.21316E-13  32.000     -0.54001E-12
73-0.20521E-03 0.60000E-02 0.00000E+00-0.39968E-13  16.000      0.00000E+00
74-0.21400E-03 0.60000E-02-0.24712E-03 0.44409E-13  64.000     -0.20384E-12
75-0.22667E-03 0.60000E-02-0.46679E-03 -49.561      -4656.0       270.56
76-0.22667E-03 0.60000E-02-0.46679E-03  49.561       4608.0      -270.56
77-0.25849E-03 0.60000E-02-0.64782E-03-0.74607E-13 -16.000     -0.16875E-12
78-0.29616E-03 0.60000E-02-0.85432E-03-0.30198E-13  48.000      0.47740E-12
79-0.28233E-03 0.60000E-02 0.00000E+00-0.12434E-13  16.000      0.00000E+00
80-0.30470E-03 0.60000E-02-0.24014E-03-0.39080E-13  48.000     -0.13767E-12
81-0.34939E-03 0.60000E-02-0.41262E-03 -557.03      -3872.0       420.80
82-0.34939E-03 0.60000E-02-0.41262E-03  557.03       3904.0      -420.80
83-0.40527E-03 0.60000E-02-0.49364E-03 0.95923E-13  64.000     -0.64926E-12
84-0.46693E-03 0.60000E-02-0.60819E-03-0.75495E-13  48.000      0.80203E-12
85-0.34344E-03 0.60000E-02 0.00000E+00 0.74607E-13  16.000      0.00000E+00
86-0.36462E-03 0.60000E-02-0.14797E-03-0.67502E-13  16.000      0.35527E-13
87-0.49756E-03 0.60000E-02-0.24649E-03 -222.14      -1680.0       347.53
88-0.49756E-03 0.60000E-02-0.24649E-03  222.14       1664.0      -347.53
89-0.80121E-03 0.60000E-02-0.27061E-03 0.63950E-13 -32.000     -0.28422E-12
90-0.92550E-03 0.60000E-02-0.37166E-03 0.24869E-13  48.000      0.35705E-12


 ---------------------------------------------------------------------------
        9999        9999          0
10 V E R A L L   T I M E   0.00
```

```
NODAL POINT INPUT               =    0.00
ELEMENT STIFFNESS FORMATION     =    0.00
NODAL LOAD INPUT                =    0.00
TOTAL STIFFNESS FORMATION       =    0.00
STATIC ANALYSIS                 =    0.00
EIGENVALUE EXTRACTION           =    0.00
FORCED RESPONSE ANALYSIS        =    0.00
RESPONSE SPECTRUM ANALYSIS      =    0.00
STEP-BY-STEP INTEGRATION        =    0.00

TOTAL SOLUTION TIME             =    0.00
```

OUTPUT FROM 'KSAP II' : [illegible]

8 node El.E02/90d3s; [illegible] MESH-[illegible].inp (10/30/87)

---STRESS OUTPUT LOCATIONS---

1 21 0 0 0 0 0 0

121 - N O D E S O L I D E L E M E N T S T R E S S

ELEMENT	LOAD	LOCATION	SIG-XX	SIG-YY	SIG-ZZ	SIG
1	1	1	[illegible]	0.107370E+04	-0.135072E+03	-0.183894E
2	1	1	-0.513450E+03	0.107229E+04	-0.142729E+03	-0.680461E
3	1	1	[illegible]	0.105201E+04	-0.192748E+03	-0.160091E
4	1	1	[illegible]	0.110929E+04	-0.617359E+02	0.205791E
5	1	1	[illegible]	0.172531E+04	0.130826E+03	-0.847874E
6	1	1	-0.354991E+03	0.172149E+04	0.122555E+03	-0.224551E
7	1	1	-0.322860E+03	0.169723E+04	0.742568E+02	-0.278600E
8	1	1	[illegible]	0.177025E+04	0.204648E+03	-0.346053E
9	1	1	-0.442210E+03	0.144015E+04	-0.648239E+02	-0.160910E
10	1	1	-0.452332E+03	0.143353E+04	-0.681815E+02	-0.545370E
11	1	1	-0.467550E+03	0.141654E+04	-0.110922E+03	-0.864481E
12	1	1	-0.427112E+03	0.146475E+04	-0.106091E+03	-0.352275E
13	1	1	-0.466043E+03	0.168614E+04	0.625469E+02	-0.809481E
14	1	1	-0.424475E+03	0.165391E+04	0.590736E+02	-0.224869E
15	1	1	-0.469036E+03	0.165933E+04	0.164611E+02	-0.312321E
16	1	1	-0.414558E+03	0.171243E+04	0.116455E+03	-0.322742E
17	1	1	0.447563E+03	0.214096E+05	-0.603122E+01	-0.126553E
18	1	1	0.440146E+03	0.214656E+05	-0.926818E+01	-0.392549E
19	1	1	0.408289E+03	0.214536E+05	-0.189085E+02	-0.660276E
20	1	1	0.227465E+03	0.214581E+05	0.876635E+01	-0.637688E
21	1	1	0.432792E+03	0.207950E+05	0.692551E+01	-0.763006E
22	1	1	0.422190E+03	0.207926E+05	0.386937E+01	-0.219859E
23	1	1	[illegible]	0.207794E+05	-0.648407E+01	-0.335985E
24	1	1	0.314506E+03	0.207436E+05	0.211307E+02	-0.384381E
25	1	1	[illegible]	0.212235E+05	0.184790E+00	-0.104566E
26	1	1	0.437833E+03	0.212341E+05	-0.135654E+00	-0.326547E
27	1	1	[illegible]	0.212204E+05	0.124554E+02	-0.518971E
28	1	1	0.134009E+03	0.211650E+05	-0.132877E+02	-0.369496E
29	1	1	0.434514E+03	0.210275E+05	0.709911E+00	-0.742767E
30	1	1	[illegible]	0.210253E+05	0.245991E+00	-0.212575E
31	1	1	[illegible]	[illegible]	0.127670E+02	-0.333745E
32	1	1	[illegible]	0.209358E+05	-0.129398E+02	-0.451993E

------ ENERGY RELEASED [illegible] Directions -----

0.24492099 [illegible] -0.23302503

===

121 - N O D E S O L I D E L E M E N T S T R E S S

ELEMENT	LOAD	LOCATION	SIG-XX	SIG-YY	SIG-ZZ	SIG
1	1	1	-0.459452E+03	[illegible]	-0.131187E+03	-0.677179E
2	1	1	-0.455305E+03	0.106068E+04	-0.118742E+03	-0.147760E
3	1	1	-0.258807E+03	0.105071E+04	-0.244673E+03	0.945992E
4	1	1	[illegible]	0.114196E+04	-0.112547E+03	0.384709E

5	1	1	-0.461091E+03	0.172665E+04	0.130259E+03	-0.313154E
6	1	1	-0.455280E+03	0.172218E+04	0.122723E+03	-0.383426E
7	1	1	-0.298793E+03	0.165999E+04	0.821536E+02	0.256255E
8	1	1	-0.595150E+02	0.168331E+04	0.272012E+03	0.265[illegible]E
9	1	1	-0.330[illegible]E+03	0.144035E+04	-0.657823E+02	-0.634614E
10	1	1	-0.339405E+03	0.145545E+04	-0.319520E+02	-0.265199E
11	1	1	-0.547864E+03	0.140766E+04	-0.172534E+03	0.148727E
12	1	1	-0.444362E+02	0.157720E+04	-0.523599E+02	0.353036E
13	1	1	-0.509716E+03	0.166601E+04	0.623993E+02	-0.264511E
14	1	1	-0.550907E+03	0.168559E+04	0.598138E+02	-0.825761E
15	1	1	-0.631592E+03	0.162604E+04	0.232740E+02	0.248813E
16	1	1	-0.497861E+03	0.158064E+04	0.179141E+03	0.628181E
17	1	1	0.454919E+03	0.214599E+05	-0.942980E+01	-0.491934E
18	1	1	0.477618E+03	0.214293E+05	0.136786E+02	-0.257382E
19	1	1	0.445130E+03	0.210855E+05	-0.246355E+02	-0.365893E
20	1	1	-0.599666E+02	0.172619E+05	-0.605275E+02	0.520656E
21	1	1	0.425392E+03	0.209036E+05	0.678539E+01	-0.255128E
22	1	1	0.415775E+03	0.208449E+05	0.590236E+01	-0.134598E
23	1	1	0.365666E+03	0.205495E+05	-0.344725E+01	0.873985E
24	1	1	0.285974E+03	0.187183E+05	0.716737E+02	0.830356E
25	1	1	0.451999E+03	0.212285E+05	-0.208313E+00	-0.390673E
26	1	1	0.471023E+03	0.212024E+05	-0.923305E+01	-0.246273E
27	1	1	0.562284E+03	0.212840E+05	0.285149E+02	-0.109388E
28	1	1	0.316123E+03	0.218731E+05	-0.228633E+01	-0.175313E
29	1	1	0.428236E+03	0.210360E+05	0.883838E+00	-0.308181E
30	1	1	0.416479E+03	0.210620E+05	0.254047E+01	-0.152305E
31	1	1	0.343974E+03	0.210059E+05	0.111353E+03	-0.202870E
32	1	1	0.141540E+03	0.202542E+05	-0.312466E+02	-0.747790E

------ ENERGY RELEASED in (x, y, z) directions -----

0.64558025 14.140609 1.3794686

===

[illegible] - N O D E S O L I D E L E M E N T S T R E S S

ELEMENT	LOAD	LOCATION	SIG-XX	SIG-YY	SIG-ZZ	SIG
1	1	1	-0.455077E+03	0.108435E+04	-0.104790E+03	-0.231523E
2	1	1	-0.312685E+03	0.106184E+04	-0.223837E+03	0.164030E
3	1	1	-0.214170E+03	0.111562E+04	-0.208921E+03	0.427663E
4	1	1	-0.108979E+03	0.121222E+04	-0.612078E+02	0.279814E
5	1	1	-0.502761E+03	0.172720E+04	0.131473E+03	0.142089E
6	1	1	-0.458382E+03	0.168180E+04	0.133735E+03	0.303849E
7	1	1	-0.293367E+03	0.157673E+04	0.115431E+03	0.354324E
8	1	1	-0.916770E+02	0.154290E+04	0.218118E+03	0.187686E
9	1	1	-0.262870E+03	0.145925E+04	-0.245710E+02	-0.867483E
10	1	1	-0.536140E+03	0.141648E+04	-0.157500E+03	0.999497E
11	1	1	0.151097E+02	0.155991E+04	-0.132885E+03	0.522352E
12	1	1	0.169432E+03	0.173573E+04	0.118579E+02	-0.291787E
13	1	1	-0.574305E+03	0.168757E+04	0.631362E+02	-0.302754E
14	1	1	-0.623795E+03	0.165204E+04	0.718584E+02	0.282842E
15	1	1	-0.614966E+03	0.149768E+04	0.481470E+02	0.791060E
16	1	1	-0.156871E+03	0.141790E+04	0.119957E+03	0.196302E
17	1	1	0.490080E+03	0.214341E+05	0.147009E+02	-0.169694E
18	1	1	0.474299E+03	0.216901E+05	-0.125472E+02	-0.286492E

19	1	1	-0.425966E+02	0.177421E+05	-0.115167E+03	0.522225E
20	1	1	-0.652782E+02	0.148759E+05	0.447546E+02	0.123125E
21	1	1	0.419860E+03	0.208458E+05	0.784515E+01	-0.848678E
22	1	1	0.402690E+03	0.205723E+05	0.107676E+02	0.112856E
23	1	1	0.353939E+03	0.136384E+05	0.434208E+02	0.102231E
24	1	1	0.462415E+02	0.164075E+05	0.722564E+01	0.767311E
25	1	1	0.483780E+03	0.212003E+05	-0.110830E+02	-0.183666E
26	1	1	0.659948E+03	0.213080E+05	0.260463E+02	-0.107525E
27	1	1	0.411701E+03	0.217331E+05	0.185557E+02	-0.153453E
28	1	1	0.493617E+02	0.216361E+05	0.333374E+02	-0.900981E
29	1	1	0.421236E+03	0.210689E+05	0.288468E+01	-0.113331E
30	1	1	0.400065E+03	0.210154E+05	-0.260228E+01	-0.158576E
31	1	1	0.384367E+03	0.205108E+05	-0.206248E+01	-0.540002E
32	1	1	0.175770E+03	0.204220E+05	-0.650713E+02	-0.976947E

S T A T I C   S O L U T I O N   T I M E   L O G

EQUATION SOLUTION = 0.00
DISPLACEMENT OUTPUT = 0.00
STRESS RECOVERY = 0.00

END
DATE
FILMED
DTIC
JULY 88